

**WIN
A \$1,500
PRINTER PACKAGE
(SEE PAGE 35)**

WE TEST 9 PRINTERS AND MODEMS FOR THE ROAD!

COMPUTE

JUNE 1992

ULTIMATE PCs!

**HOTTEST MACHINES
FOR WRITING, PUBLISHING,
AND ACCOUNTING**

ULTIMATE UTILITIES!

• **NORTON vs. PC TOOLS**

ULTIMATE GAMES!

• **SUPER SIMULATORS**

PLUS

• **POWER PIMs**

• **PC HEALTH**

• **MULTIMEDIA**

SPECIAL

PULL-OUT SECTION!

• **DESKTOP GRAPHICS**



0 71486 02193 3



64/128 VIEW

*Gazette needs 64 and 128 programs every month.
If you are a programmer, here are
some tips that can help you make a sale.*

Tom Netsel

Gazette wants to buy your 64 and 128 programs. Sell just one program, and you can easily recoup the cost of your entire computer system.

We rely on our readers for the programs we need each month to fill Gazette and Gazette Disk. Here are some tips on what we want and ways to make it easier for you to make a sale.

Send your program and documentation on a disk. Save each twice, in case one gets damaged. Send a printout of your documentation and a short cover letter explaining what your program does and how to run it. Enclose a daytime telephone number. Don't send a printout of your program listing. Send postage if you want your material returned.

To increase your chances of making a sale, do all you can to make it easy for us to use and understand your program. I've rejected programs because authors have made my job tougher.

Here's what we like to see in a submission. First of all, use the correct address. Don't delay things by writing to a post office box number we haven't used in years. Send programs to Gazette Submissions Reviewer, COMPUTE, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408.

Deadlines, interruptions, and weak coffee can put me in a bad mood at submission meetings. Imagine that I've a stack of programs to get through, and I've just opened yours. You've got 60 seconds to impress me.

You'll be off to a good start with that note that explains what your program

does and how to run it. If there are numerous files, backups, and demos on your disk, I can get very annoyed deciphering cryptic filenames, trying to decide which file to load first.

If we've had your package open for more than a minute and we still don't know what it's supposed to do, we'll do one of two things. We'll either reject the program outright or toss it back in the pile until next month. In the latter case, we don't totally reject it, but we're not jumping to buy it either. This bumping process can go on for months.

With your letter, send a printout of your documentation. Here's where you can really boost your chances for a sale. Write in the style we use in the magazine! Explain what your program does, why it might be useful or entertaining, and whether it's in BASIC or machine language. Then explain how to use the program. Many authors fail to do this.

It's very important that you send documentation on disk as well as paper. Save it as a PETSCII, ASCII, or SpeedScript-compatible file. DO NOT USE GEOS! I load your documentation into my 128, edit it, convert it to ASCII, and upload it to an IBM for additional editing and typesetting. Retyping your documentation wastes time. If I have a program whose instructions need minor editing and one that requires hours of work, guess which one I'll buy?

Finally, tell us about yourself: your occupation, age, hobbies, and so on. We like to publish a little information about our authors. □

GAZETTE

64/128 VIEW

G-1

If you send a game or utility to Gazette, follow these tips to boost your chances for a sale.
By Tom Netsel.

FEEDBACK

G-2

Questions and comments from our readers.

RELATIVELY SIMPLE

G-6

Many programmers shy away from using relative files, but let's take another look at these black sheep of the file family.
By Jayson Johnson.

REVIEWS

G-12

Tie Break, Calc II, and Bad Blood.

BEGINNER BASIC

G-18

Turn keyboard characters into movable sprites.
By Larry Cotton.

MACHINE LANGUAGE

G-20

Use a rotating buffer to determine a program's starting address and more.
By Jim Butterfield.

WORLD VIEW

G-21

Czechoslovakia loves its Commies, but politics has nothing to do with it.
By Emil Heyrovsky.

GEOS

G-22

GEOS graphics require a good management system.
By Steve Vander Ark.

PROGRAMMER'S PAGE

G-24

Here's a handy list of POKes, WAITs, and SYS calls.
By Randy Thompson.

PROGRAMS

File Logger	G-25
Demo Maker	G-26
F/X Plot 128	G-30
Sprint III	G-33
Pad Design	G-38

Address Correction

In "Commodore Clips" (February/March 1992) we listed the wrong post office box number for Clip Art Cupboard. The correct address is P.O. Box 317774, Cincinnati, Ohio 45231. We regret the error.

Amortization Table

I have been trying in vain to get hold of a program that will list amortization tables with an option of payments every two weeks. Can anyone help me?

MARGIT DES LAURIERS
SANDSPIT, BC
CANADA

In the May issue, Larry Cotton presented an amortization program in his "Beginner BASIC" column that displays payments on a monthly or yearly basis. For payments other than monthly, however, you might try the following program. It asks for the amount borrowed and the number of payments per year. If you wish to make payments every two weeks, enter 26 at the second prompt. It will then request the annual interest rate and the duration of the loan in years.

The program will calculate the payment you must make for each period and then print a table showing the breakdown of interest and principal on each payment and the remaining balance. At the end of a year, it will print a total of the principal and interest paid. The program will pause and wait for you to press any key before printing the next year's schedule of payments.

```
RM 50 KS="PRESS ANY KEY
      {SPACE}TO CONTINUE
      "
GR 100 PRINT"{CLR}
SE 110 DEF FNA(X)=INT(X*
      100+.5)/100
BJ 120 INPUT"AMOUNT BORR
      OWED";E
MB 130 INPUT"NUMBER OF P
```

```
      AYMENTS PER YEAR"
      ;N
AM 140 INPUT"ANNUAL INTE
      REST RATE";I:I=I/
      100
AB 150 INPUT"DURATION IN
      YEARS";D
CQ 160 R=(I*E/N)/(1-1/(I
      /N+1)^(N*D))
AC 170 PRINT
JX 180 PRINT"YOUR PERIOD
      IC PAYMENT WILL B
      E $";FNA(R)
BE 190 PRINTKS
QX 195 GET WS:IF WS=""TH
      EN 195
HD 200 PRINT
AF 210 AT=0:A2=0:EN=E:RT
      =0:IT=0:B=0:D1=N
KJ 220 IF INT(D)>=1THEN2
      40
AJ 230 B=B+1
JD 240 FORB=1TO INT(D)
RK 250 PRINT CHR$(147)"
      {DOWN}AMORTIZATIO
      N SCHEDULE";TAB(2
      5);" YEAR #";B
PF 260 PRINT:PRINT" #TA
      B(5)"INTEREST";TAB
      (17)"PRINCIPAL";
JK 270 PRINT TAB(31)"BAL
      ANCE":PRINT
PM 280 FORB1=1TOD1
JA 290 IV=FNA(EN)*I/N
CJ 300 RT=RT+1:A=R-IV:AT
      =AT+A:EN=E-AT
FS 310 IFRT<>N*DTNEN330
DX 320 R=R+EN:A=A+EN:AT=
      AT+EN:EN=0
HE 330 I2=I2+IV:IT=IT+IV
      :A2=A2+A
AJ 340 A2=INT(A2*100+.5)
      /100
MX 350 PRINTB1;TAB(4);FN
      A(IV);TAB(16);FNA
      (A);
XQ 360 PRINT TAB(30);FNA
      (EN)
KK 370 NEXT
FD 380 IF RT<>N*DTNEN400
SF 390 PRINT:PRINT"LAST
      {SPACE}PAYMENT";F
      NA(R):PRINT
RP 400 PRINT:PRINT"FOR T
      HE CURRENT YEAR Y
      OU PAID"
DB 410 PRINT"$";FNA(IT);
      "IN INTEREST "
HH 420 PRINT"AND $";FNA(
      A2);"IN PRINCIPAL
      "
FP 430 IF B=D OR B>D THE
      N 540
CM 440 PRINT:PRINTKS
SF 450 GET WS:IF WS=""TH
      EN 450
SP 460 PRINTCHR$(147)
KG 470 IT=0:A2=0
PA 480 NEXT
BM 490 B=B-1
FP 500 IF D=BTHEN540
HR 510 D1=((D-INT(D))*12
```

```
      )/12*N
CM 520 B=B+1
RF 530 GOTO 250
GP 540 PRINTKS
ER 550 GET WS:IF WS=""TH
      EN 550
EK 560 END
```

Genealogy Programs

I am looking for a family tree program for my 64. Can you help?

FREDERICK J. CARLETON
METAIRIE, LA

We published "Climbing Your Family Tree" in the February 1991 issue of *COMPUTE* in which the author described several programs available for the 64 that let users enter family data. You might want to contact the following companies about their genealogy programs. Here's a list of the ones mentioned in the article.

Family—\$34.95

PFA

8600 Old Spanish Tr., Ste. 79
Tucson, AZ 85710
(800) 366-1372

PED C and FGS—\$39.95

BYTEWARE

906 West 6th Ave.
Monmouth, IL 61462
(309) 734-7096

Arbor-Aide—\$34.95

SOFTWARE SOLUTIONS

7378 Zurawski Ct.
Custer, WI 54423
(715) 592-3594

Family Roots—\$225.00

QUINSEPT

P.O. Box 216
Lexington, MA 02173
(800) 637-7668

Keyboard Wanted

The time has come for me to locate a new detached keyboard for my 128D. Perhaps one of your readers may have one for sale.

STEPHEN VAN EGMOND
360 FRONT RD.
LASALLE, ON
CANADA N9J 1Z5

Questions
and answers about
amortization
tables, genealogy
programs
for the 64/128,
and more

Stay On the Fast Track with TENEX!

Always the most value for your money!

**Best Deal Ever on
the Amiga 500!**

Call for details!

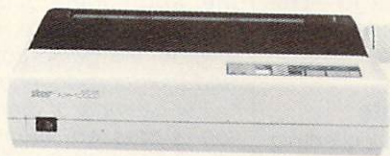
**Commodore®
AMIGA®**



**Commodore-Ready Printer
from Star
At An Incredible Price!**

\$119⁹⁵

star
THE COMPUTER PRINTER



NX-1000C.

Enjoy near-letter quality at 36 cps and high-speed draft at 144 cps. Features the new paper parking function and convenient front panel controls. Includes four built-in fonts and over 20 typestyles. This is the Commodore-ready version of the NX-1000. You can just plug it in and start printing—no additional interfaces or cables are required.

NX-1000C Printer	75060	\$299.00
		\$179.95
		SPECIAL! \$119.95
Ribbon (NX-1000C)	75471	\$5.95
Dust cover	77789	\$8.95

OTHER STAR PRINTERS

NX-1001 Multi-Font	90895	\$139.95
NX-2420 Rainbow	A51047	\$299.95
NX-1020 Rainbow	A51027	\$179.95
LaserPrinter 4	A57934	\$879.00

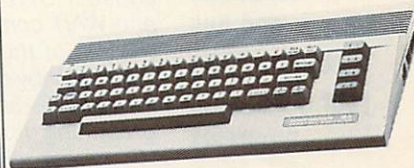
Don't miss out on the lowest prices on the Amiga 500, plus a full line of Commodore and Amiga hardware, software, and accessories.

Call today to receive your **FREE** catalog with the greatest prices on the most popular hardware and software!



**Commodore 64
Computer**

**Only
\$139⁹⁵**



**Commodore 1541 II
Disk Drive**

**Only
\$169⁹⁵**



TENEX™
Computer Express

Order Today! Call 1-800-PROMPT-1

P.O. Box 6578
South Bend, IN 46660
(219)259-7051 FAX (219)259-0300
We gladly accept mail orders!
Circle Reader Service Number 170



Shipping, Handling, Insurance

Order Amount	Charge
less than \$19.99	\$4.50
\$20.00-\$39.99	\$5.75
\$40.00-\$74.99	\$6.75
\$75.00-\$99.99	\$7.75
\$100.00-\$149.99	\$8.75
\$150.00-\$299.99	\$9.75
\$300.00-\$499.00	\$11.75
\$500.00-\$699.99	\$18.75
\$700.00-\$999.99	\$24.75
\$1000 & Over	2.5% of Order

COMMODORE 64 and COMMODORE 128 are registered trademarks of Commodore Electronics, Ltd. AMIGA is a registered trademark of Commodore Amiga Inc., NOTE: Due to publishing lead-times, product prices and specifications are subject to change without notice. *APO, FPO, AK, HI, CN, VI, GU, and foreign orders are subject to additional shipping charges.

Circle Reader Service Number 170

G5W

Cursor SYS for 128

SYS 52591 will turn on a cursor on the 128 in 80-column mode. But what is the SYS number to release the cursor?

ROBERT NELLIST
BROCKPORT, NY

A SYS to address 52591 (\$CD6F) will turn the cursor on in either 40- or 80-column display mode. To disable the cursor in either mode, enter SYS 52639 (\$CD9F).

If you do any programming in 64 mode, you might be interested in reading Randy Thompson's "Programmer's Page" this month. His column contains a handy reference table for 64 programmers that lists dozens of interesting and unusual SYS calls, POKES, and WAIT commands.

Most of these items were submitted by our readers. We encourage—and pay for—programming tips used on the "Programmer's Page."

Sports Card Update

I have just started collecting trading cards, so Kevin Davis's Sports Card Collector (November 1991) is a blessing. One thing that I felt was missing was an indication of when you had last updated the prices. So I added the following to the program, and it works great.

```
195 PRINT"([12 SPACES])UPDATE:
";DAS$
390 INPUT#8,YT:INPUT#8,DAS$:
INPUT#8,GC
615 INPUT"[<8>]ENTER DATE:
MO. DA. YR.(whl)";DAS$
1240 PRINT#8,YT:PRINT#8,DAS$:
PRINT#8,GC
```

DOUGLAS JEFFERY
TELKWA, BC
CANADA

Writing Games

I am writing a game on my 64 in BASIC. It's getting a little complicated, and I'm beginning to think that maybe I

should be writing it in machine language. How can I have a sound track running, sprites moving, and the computer awaiting input all at the same time?

BRYAN PEASE
LIVERPOOL, NY

Computer games may seem to do many things at once, but they're actually doing only one thing at a time. Computers follow instructions sequentially. A system that does several different things at once is possible, but you would need more than one computer or microprocessor (each operating sequentially). To give your program the appearance of simultaneous action, you need to plan. Separate the actions into subroutines. IF-THEN can decide whether or not you want to gosub to the appropriate routine. It might help to write, in plain English, the game conditions and their consequences.

For example, if the fire button is pressed, then launch missile and set missile flag. If the joystick moves, then move ship sprite. If one second has passed, then play another note of the song. If the missile flag is set, then move missile sprite again. Repeat the above loop.

First, you check for the joystick button. If it's pressed, then gosub to the appropriate routine. If it's not, forget about launching the missile until the next time through the loop. Once you've launched the missile, you want it to continue moving, which is the reason for the missile flag. Whether or not the button was down, you next peek the joystick to see if the player wants to move and update the ship's position. Third, you check the jiffy clock by reading variable TI or TIS to see how much time has gone by. If a second (or whatever time

period you've chosen) has passed, play the next note of the song.

Next, move the missile sprite if the flag is set, and go back. The program loops around, checking the joystick twice, checking the time, checking a variable, and taking any necessary action.

The computer works quickly, so individual actions seem to happen simultaneously.

To convert the above outline into a playable game, you'd need a few more subroutines. One would check the collision register in case the missile has hit something. Another would erase the sprite and reset the flag as soon as the missile has reached the top of the screen. And, of course, you'd have to translate the outline into BASIC code.

There's another technique that's even closer to simultaneous action, but it requires from an intermediate to an advanced knowledge of machine language. Sixty times a second the computer stops what it's doing and takes some time to redraw the image on the screen. The main program is being constantly interrupted. Using a wedge, you can divert the interrupt to your own machine language program, which could play music, move sprites, or whatever you choose. Such interrupt-driven routines are sometimes difficult to implement, but they can be very effective.

Do you have a question or problem? Have you discovered something that could help other 64/128 users? Do you have a comment about something you've read in Gazette? We want to hear from you. Send your questions and comments to Gazette Feedback, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408. □

An update
to a sports card
collection
program and tips
about writing
your own game
programs

VIDEOFOX

The Tool For Creative Video Buffs

- Generate video titles, opening credits, window advertising, animation or other small trick movies
- All of these exciting effects are easy and fun for you to do with our new Videofox software
- Provides 18 special effects such as scrolling, combing, windshield wiper and spiral mixing
- Mix text, graphics and effects to produce hundreds of combinations
- Independent adjustment of foreground and background colors
- Page flipping in real time for perfect animation sequences

ONLY \$59.95

VIDEO DIGITIZER

- Digitize black and white or color pictures
- Digitize any video source including VCR
- Digitize either 4, 7 or 13 level grey levels
- Menu controlled picture brightness
- Includes three independent software programs for total control and editing of digitized images: DIGISON ~ DIGIFOX ~ DIGIMULTI
- Free color filters included for digitizing color images from black and white cameras
- separate adjustment of brightness levels for each of the red - green - blue primary colors

HANDYSCANNER 64

The Worlds First Handscanner for the 64!

- Professional quality super high 400 dots per inch resolution ~ Reads the graphics from any printed document
- Converts any material to digitized graphics in seconds ~ B/W setting for crisp reproduction of high contrast line art
- Elaborate grey-tone scale digitizes color or black & white photos using 3 built in dithering settings
- Enlarge or reduce 300% to 33% ~ Graphic memory of 640 X 400 standard (640 X 800 with Pagefox module)
- Included software has all the standard functions of a good drawing program

ONLY \$299.95

PAGEFOX

3 Easy To Use Editors For Perfect Home Desktop Publishing

GRAPHIC EDITOR ~ TEXT EDITOR ~ LAYOUT EDITOR

- Completely menu driven
- 100Kb storage enlargement module keeps entire page in memory
- Uses proportional mouse or joystick for total control over text, graphics or picture

ONLY \$139.95

MIDI 64 - Only \$49.99

- Full specification MIDI at a realistic price
- MIDI In - MIDI Out - MIDI Thru
- Works with Sampler and Adv. Music System
- MIDI CABLES (4 ft. prof. quality) - Only \$*8.99
- FREE cables when you buy MIDI & ADV. MUSIC at same time

DIGITAL SOUND SAMPLER Only - \$89.99

THE ADVANCED OCP ART STUDIO COMPREHENSIVE, USER FRIENDLY ART AND DESIGN SOFTWARE Only - \$29.99

ACTION REPLAY V 6.0

THE ULTIMATE UTILITY/BACKUP CARTRIDGE FOR THE C64/128

Allows You To Freeze The Action Of Any Memory Resident Program And Make A Complete Backup To Disk

WARP 25 - The worlds fastest disk serial Turbo

- Typical backup will reload in under 5 seconds
- No special formats-save directly into Warp mode
- Warp Save/Load available straight from BASIC
- RAMLOADER - Loads most commercial originals 25 times faster than normal!

UNIQUE CODE CRACKER MONITOR -

- Full monitor features
- See the code in its Frozen state not Reset state

MORE UNIQUE FEATURES - Menu driven operation

- Simple operation: Just press a button at any point
- All backups reload WITHOUT cartridge at Warp speed
- Sprite killer: Make yourself invincible-disable collisions
- Freeze HiRez screen & save in Koala & Paddles format
- Print out any screen in 16 grey scales
- 100% compatible with ALL drives and computers
- Disk utilities: fast format, directory, list and many other commands operated directly from function keys

ADV. MUSIC SYSTEM

Powerful modular program for creating, editing, playing and printing out music

- Playback thru internal sound or external MIDI keyboard/synthesiser
- Print music in proper musical notation together with lyrics using PRINTER module
- Enter music a note at a time in written music format using the EDITOR or via on screen piano KEYBOARD emulator or via an externally connected MIDI keyboard
- Generate almost unlimited sounds with the flexible SYNTHESIZER module
- Linker joins files to form large compositions

Only - \$29.99

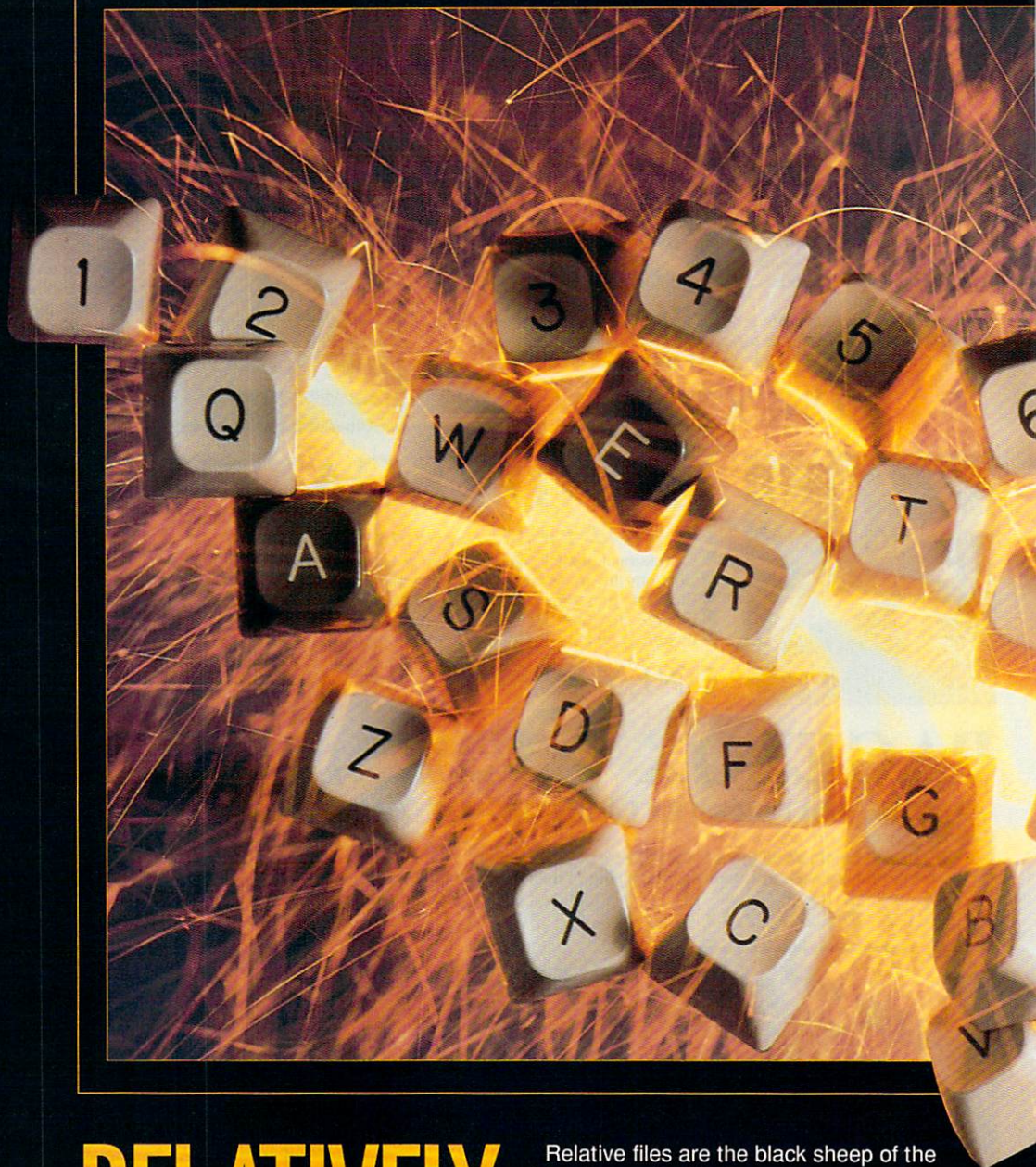
MAKE THE MOST OF YOUR ACTION REPLAY

GRAPHICS SUPPORT DISK

- View screens in a slide show sequence
 - Add scrolling messages to your saved screens
 - Contains full sprite editor
 - Explodes sections of saved screens to full size
- Only - \$19.99**

SUPERCUNCHER - ONLY \$9.99

Turn your Action Replay into a super powerful program compactor. Reduce programs by up to 50%! Further compact programs already crunched by Action Replays compactor



RELATIVELY SIMPLE

PROGRAMMERS
OFTEN SHY AWAY FROM
RELATIVE FILES, BUT
MAYBE IT'S TIME TO TAKE
ANOTHER LOOK
AT THIS ALTERNATE METHOD
OF DATA STORAGE.

BY JAYSON JOHNSON

Relative files are the black sheep of the data storage family. Programmers may try them once, have a few problems, and then give up on them. Then they spread the word to their friends to stay away from relative files. The reputation spreads. Unlike sequential files, relative files get no respect, but maybe it's time for a change.

Relative files do have good points and advantages. For one thing, they offer far more speed and efficiency than other methods of data storage. Since they can access data without reading through every file that precedes the desired information, relative files offer almost random access. If you have a large sequential file, you must first load the entire file into the computer's memory for processing. Then to retrieve data, you may have to read every file before you find the information you want. With sequential files, if you want a



piece of information that's stored in record number 100, you'll have to plow through records 1-99 first.

Suppose you have a CD collection of 500 titles and you want to catalogue it on index cards. If you put the titles in alphabetical order and read them as a sequential file reads data, you'd have to flip through the whole stack before you came to anything by ZZ Top or Pia Zadora.

With relative files, you can go directly to the data you want and read only that data. Just as with index cards, you can home in on the information you want without starting from the beginning and flipping sequentially through every file.

Since a relative file program uses only enough memory to read desired data, very little computer memory is actually used at one time. A small program can access 163K of information that could be stored on a single-sided disk.

Follow the Rules

Creating and using relative files is fairly easy, providing you follow a few simple rules. First, let's get some terminology straight. With relative files, information is stored in records, and each record contains various fields. In the following example, we'll create a mailing list. This whole list is our file. Each person on our list will be a record, and the various pieces of data about each person will be our fields.

For example, each record on our list will contain a person's first and last names, street address or post office box number, city, state, ZIP or postal code, and telephone number. These are the fields we'll set up for each record. We'll begin by calculating the total size of each record. This is done by adding the number of characters in each field. With our mailing list, we'll estimate how many characters each field should contain.

Field#	Field name	Number of Characters
1	Last name	15
2	First name	15
3	Address1	20
4	Address2	20
5	Address3	20
6	State	10
7	Zip Code	09
8	Phone #	12

Total 121

The record size for this file would be 121. Since the PRINT# statement used with relative files adds a carriage return to the end of each string, you should add one extra character to each field. So make it 129. If you try to write 15 characters plus a CHR\$(13) (carriage return) to a record set up for 15, you'd get an ERROR 51, OVERFLOW IN RECORD.

DOS uses a single ASCII character to represent the record length. This means that the largest single record could be 254 characters. This is very important to remember. If you want more information stored in each record, you'll have to split them in two. For example, a 500-character record could be split into two 250-character records. You could use odd and even numbers to identify each data pair.

The largest number of records you can have is 65,535, but you'll never use this many because of space limitations on a floppy disk. A freshly formatted disk should show 644 blocks free. Blocks, or sectors, are the areas on a disk that can hold up to 256 characters of information. DOS uses 2 of these characters, so that leaves 254 bytes available for data.

Relative files use side sectors as an index to keep track of which sectors contain data. Each side sector has room for 120 two-block pointers that let DOS quickly locate the various blocks assigned to the file.

As many as six side sectors can be assigned to a file since each can deal with 120 disk sectors (not records) for a total of 720, more than the number of blocks on a disk. Filling a 664-block disk would use 6 blocks for side sectors, leaving 658 blocks for data. Each block can hold 254 characters, giving you 167,132 characters in the largest possible relative file. You can split this up any way that is convenient for your data. You could have 1671 records of 100 characters each or 658 records of 254 characters each.

Create a File

We have to create a file before records can be stored or read. This is done by opening a data channel with the de-

sired filename and record length. You should note that DOS won't let you create relative files with record lengths of 42, 58, or 63. These numbers represent ASCII values that have special meaning to DOS. The format for creating a relative file is as follows.

OPEN file#, device#, channel#, "file-name,L," + CHR\$(record length)

Let's start writing a program and call our file MAILLIST. We previously counted the number of characters we'd need in each record and decided on 129. Type in the following OPEN command, but don't run the program until we complete it.

**1000 REM OPEN MAILLIST FILE
1005 OPEN 1,8,2,"MAILLIST,L,"+CHR\$(129)**

Remember not to run it yet. It's now possible for us to put information into the file. In DOS, there's no command to check the number of records in a relative file; however, it's possible to store this number manually within the first record. Let's do this by positioning the record pointer to the first record and storing the number 1. To do this, we must open the command channel.

1010 OPEN 15,8,15

The POSITION command for record 1, position 1 is

1015 PRINT#15,"P"+CHR\$(2)+CHR\$(1)+CHR\$(0)+CHR\$(1)

The P is our pointer, and it tells the drive to look for a certain record. In CHR\$(2) above, 2 is the channel we opened when we created the file in line 1005 (OPEN 1,8,2.). Next, put the number 1 into the record, because all we have now is one record in our file.

1020 PRINT#1,1

Print to the file number we assigned when we opened the file. (Remember OPEN 1,8,2.) Then, close the file and the command channel, and the file is ready to use.

1030 CLOSE1:CLOSE15

Run the program now, and a file called "MAILLIST" will be created and placed in the disk's directory.

Writing and Reading

To read from or write to any record in our file, we open two channels, the DOS command channel and a file channel. Now we can position the record pointer to any record and field in the

file with the DOS POSITION command.

PRINT#15,"P"+CHR\$(channel number)+CHR\$(rec# lo)+CHR\$(rec# hi)+CHR\$(position within record)

Notice that in the above line there are two characters that represent the record number (rec# lo and rec# hi). The formula for calculating these two numbers is as follows.

RECORD # = REC HI * 256 + REC LO

The record number can also be determined in other ways.

**REC HI = INT(RECORD #/256)
REC LO = RECORD # - REC HI*256**

Translated, the command sequence would be as follows. (You don't have to

TIPS ON USING RELATIVE FILES

Relative files are good ways to store data for quick and easy access, but many programmers avoid using them. Here are a few things to remember to avoid problems.

1. Always initialize the disk drive at the beginning of your program to reset all drive channels.
2. Always set the file pointer to the first byte in a record. The last parameter in the P command should be CHR\$(1).
3. Never write an empty string to the disk. This will cause the disk drive to lock in a continuous loop.
4. Never leave a file open after entering or editing data. Be sure to open, read/write, and close the file in the same command sequence.

If you attempt to create a file that's too large for the space remaining on your disk, you'll get ERROR 52, FILE TOO LARGE. If you plan to store a lot of data, it's a good idea to devote a whole disk to a relative file.

Here's a short BASIC program to determine the maximum number of records you can have on a disk. To use the program, you must know how many free blocks remain on your disk and the total number of characters allotted for each record.

```

10 INPUT "NUMBER OF FREE BLOCKS ON DISK";FB
20 INPUT "TOTAL CHARACTERS IN RECORD";RL: IF RL<1 OR RL>254 THEN 20
30 IF RL=42 OR RL=58 OR RL=63 THEN PRINT "DO NOT USE";RL; "CHARACTERS": GOTO 20
40 SS%=((FB/120)+.99): BA=FB-SS%
50 RP=(INT(BA*254/RL): IF RP<65536 THEN RP=65536
60 PRINT "THERE'S SPACE FOR";RP;"RECORDS"
```


INTRODUCING

COMPUTE NET



- About COMPUTE/NET
- Product Ordering
- Feedback Board
- Coming Events
- Monthly Contest



Welcome to the grand opening of COMPUTE/NET. A wealth of information awaits you. Back issues of COMPUTE, hard-to-find computer books, super software, dazzling pictures, challenging games, prizes, a complete bulletin board, and much more are here. You can even talk to the editors and authors of the magazine. Lots of surprises are planned, so keep your eyes on us.

FIND US ON Q-LINK



FREE Q-LINK STARTER KIT. FREE TIME. ORDER TODAY!

Just call our toll-free number or return the coupon, and we'll send you the Q-Link Starter Kit and software free, waive your first month's membership fee, and credit you with one hour of "Plus" time to try the service. Your \$9.95 monthly fee gives you unlimited access to all of our "Basic" services online, including a searchable encyclopedia, AND one free*hour of "Plus" services. After your free hour, you'll pay only \$4.80/hour—just 8 cents per minute—for additional use of the service.

Q-Link is a registered service mark of Quantum Computer Services, Inc.

*Long-distance charges may apply. Surcharges apply if you are a resident of Alaska, Hawaii, or Canada. Allow four to six weeks for delivery.

☐ **YES!** Send me my FREE Q-Link software, waive my first month's membership fee, and credit me with one FREE* hour of Plus time to explore the service and try COMPUTE/NET.

Name _____
Address _____
City _____ State _____ Zip _____
Home Phone _____



Use of Q-Link requires a VISA, MasterCard, or checking account.

MAIL TO

Q-Link

8619 Westwood Center Drive
Vienna, Virginia 22182-9897

Call 1-800-782-2278, Ext. 2414 today

enter NEW before typing this next section. It and the remainder of the code are meant to be added to the previous program lines.)

```

10 OPEN15,8,15
15 OPEN1,8,2,"MAILLIST": REM WE ARE
  OPENING, NOT CREATING THE FILE
40 R = 1: P = 1: REM RECORD NUMBER 1
  POSITION 1
45 GOSUB 500
90 CLOSE1: CLOSE15: END
500 REM POSITION RECORD POINTER
510 RC = R + 1: REM ADD 1 TO RECORD
  (REC1 IS ALREADY USED TO HOLD FILE
  LENGTH)
520 RH% = RC/256: RL = RC-RH% * 256:
  REM TWO-BYTE ADDRESS; RH% IS AN
  INTEGER
530 PRINT#15,"P" + CHR$(2) + CHR$(RL) +
  CHR$(RH%) + CHR$(P)
540 RETURN
999 END

```

Don't run the program yet.

Record Contents

Each record is actually a string of information in itself. To read each field separately, we must calculate where it begins within the record. Here are the record sizes we established earlier. We can use these figures to determine

where to look for each field.

Last name	1-15
First name	16-30
Address1	31-50
Address2	51-70
Address3	71-90
State	91-100
Zip Code	101-109
Phone #	110-121

Now let's make a list of pointers for each field.

```

5000 DATA 1, LNAME, 16, FNAME, 31, AD1,
  51, AD2, 71, AD3, 91, STATE, 101, ZIP,
  110, PHNO

```

As you can see, this translates into Last name at position 1, First name at position 16, Address1 at position 31, and so on.

Writing Data

Let's enter some information into a sample record. As we do, we'll write over line 45 in the earlier subroutine.

```

35 REM STORE RECORD
45 F$(1)="JOHN": F$(2)="DOE"
50 F$(3)="15 HOLLYWOOD AVENUE"
55 F$(4)="(SPACE)"
60 F$(5)="HOLLYWOOD"

```

```

65 F$(6)="CALIF."
70 F$(7)="90135"
75 F$(8)="555-964-6652"
80 RESTORE: REM RESET DATA POINTER
85 FORLP=1TO8: READP,X$: GOSUB500:
  PRINT#1, F$(LP): NEXTLP

```

When you have this final block of code entered, you may run the program typed in so far. You may also want to save it to disk.

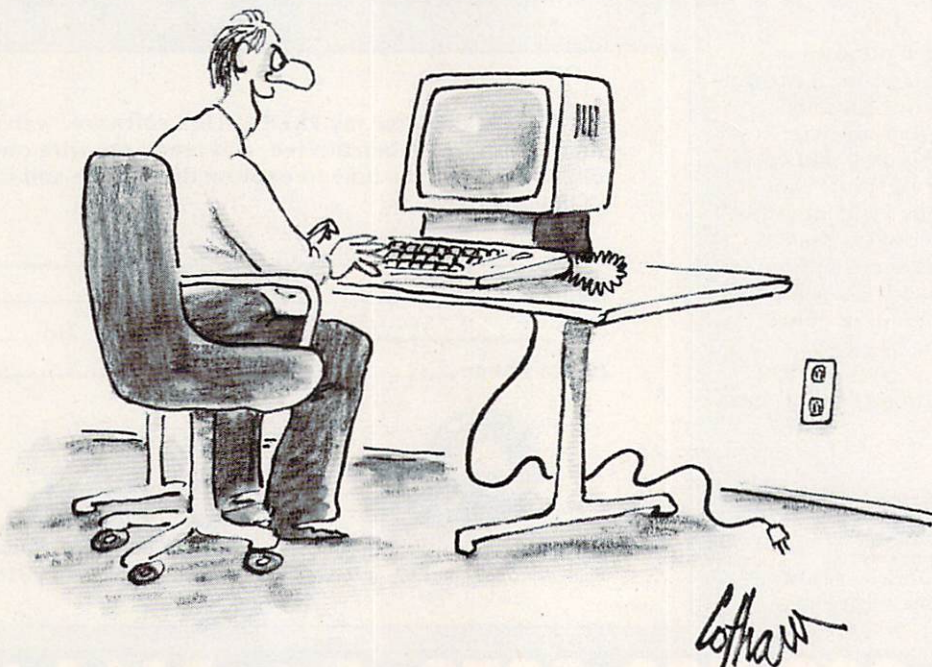
Reading Data

Reading information is just the opposite. You set the pointer and use INPUT# to read the information instead of PRINT#, which is used to write it.

```

20 GOT0100
100 REM READ RECORD
110 R=1: REM RECORD 1
115 RESTORE: RESET BASIC DATA POINTER
120 FORLP=1TO8: READP,X$: GOSUB500:
  INPUT#1,F$(LP): NEXTLP
125 CLOSE1: CLOSE15
130 PRINT"LAST NAME: ";F$(1)
135 PRINT"FIRST NAME: ";F$(2)
140 PRINT"ADDRESS: "
145 PRINTF$(3)
150 PRINTF$(4)
155 PRINTF$(5);", ";F$(6),F$(7)
160 PRINT"PHONE#: ";F$(8)
175 END

```



"You're not quite as smart as you thought you were, eh? I've unplugged you!"

GAZETTE DISK LIBRARY

VALUE-PACKED SOFTWARE AT AFFORDABLE PRICES

All Gazette disks are menu-driven for ease of use—and they feature complete documentation. Just load and you're ready to go!

SpeedScript

\$11.95

COMPUTE Publications' most popular program ever. Powerful word processing package includes SpeedScript for the 64, SpeedScript 128, spelling checkers for both 64 and 128 versions, plus an additional dozen support programs, including mail-merge and word-count utilities.

Gazette Index

\$7.95

Every article and department from Gazette—July 1983 through December 1989 issues—is indexed: features, games, reviews, programming, "Bug-Swatter," "Feedback," and the other columns. Disk features pull-down menus, help screens, superfast searching/sorting capabilities, and much more.

Best Gazette Games

\$9.95

Best dozen arcade and strategy games ever published in Gazette all on one disk. All games for Commodore 64. Titles: Crossroads II: Pandemonium, Basketball Sam & Ed, Delta War, Heat Seeker, Omicron, Powerball, Q-Bird, Trap, Arcade Volleyball, Mosaic, Power Poker, and Scorpion II.

Gazette's Power Tools

\$9.95

Fourteen of the most important utilities for the 64 ever published in Gazette. For serious users. Titles: MetaBASIC, Disk Rapid Transit, Mob Maker, Ultrafont+, Quick!, Disk Editor, Basically Music, PrintScreen, 1526 PrintScreen, Fast Assembler, Smart Disassembler, Comparator, Sprint II, and Turbo Format.

The GEOS Collection

\$11.95

Gazette's best 13 programs for GEOS and GEOS 128 users. Selection includes utilities, applications, and games. Titles: Super Printer Driver, Skeet, File Saver, Help Pad, Word Count, Directory Printer, Quick Clock, SlideShow, File Retriever, Screen Dumper, Font Grabber, GeoPuzzle, and GeoConverter.

128 Classics

\$11.95

Thirteen of Gazette's best 128 programs, including utilities, games, and applications. Titles: MetaBASIC 128, RAMDisk 128, 80-Column Disk Sector Editor, MultiSort, Block Out, Miami Ice, The Animals' Show, Cribbage, XPressCard, Sound Designer, Video Slide Show, Math Graphics, and 3-D BarGrapher.

SPECIAL OFFER!

All 6 DISKS FOR ONLY \$49.95!

A \$13.00 SAVINGS!

All prices include shipping & handling.

SpeedScript	☐ \$11.95
Gazette Index	☐ \$ 7.95
Best Gazette Games	☐ \$ 9.95
Gazette's Power Tools	☐ \$ 9.95
The GEOS Collection	☐ \$11.95
128 Classics	☐ \$11.95
Special 6-Disk Offer	☐ \$49.95

Subtotal _____

Tax* _____

Outside U.S. or Canada** _____

Total _____

Name _____

Address _____

City _____ State _____ ZIP _____

Amount enclosed \$ _____ Method of payment ☐ Check or Money Order

☐ VISA or MasterCard

Mail to Gazette Disks
324 W. Wendover Ave., Ste. 200
Greensboro, NC 27408

Credit card no. _____ Exp. date _____

Signature (required) _____

Daytime phone number _____

* Residents of North Carolina and New York add appropriate sales tax. Canadian orders, add 7% goods and services tax.

** For delivery outside the U.S. or Canada, add \$1 for surface mail or \$3 for airmail. All orders must be in U.S. funds drawn on a U.S. bank.

TIE BREAK

Tennis anyone? You don't have to be a top seed tennis professional at Wimbledon to enjoy playing DigiTek Software's Tie Break. This computerized tennis game is for tennis lovers of all levels.

Tie Break offers three modes of play: world tournament, tournament, and training. World tournament is for those who want a real challenge. Here, you select up to 16 players (human or computer), and you must defeat all of them to win the championship. You then select the world tournament that you want to enter. These vary in level of difficulty and include Wimbledon, the French Open, the Masters, the U.S. Open, the Davis Cup, and the Australian Open. For extra realism, each player is assigned a racket of a certain tension and weight.

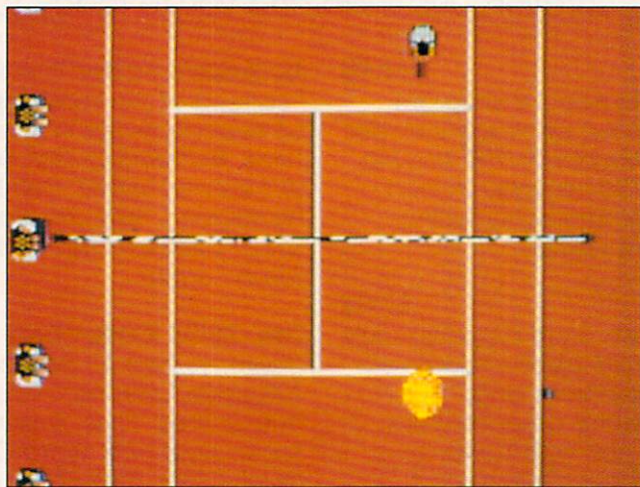
Perhaps you don't feel ready for world tournament play just yet. If you just want to compete against a few of your friends, select tournament play. Once again, each player is assigned racket weight and tension. You also select the surface type and color that you want to play on. Court types vary in difficulty since the ball may move faster or bounce differently on certain surfaces. For example, the ball will move faster on an asphalt or a PVC court than it will on clay.

Let's say that you haven't played for a while and are a little rusty. If you feel like practicing either before entering a world tournament or playing against your best friend who happens to be a tennis pro, then select training. Here, you can practice with a variety of computer players of varying skill levels.

If your friend wants to practice as well, you both can

play against each other or play doubles with two computer players. (To play against a friend, one person must use the joystick, and the other must use the 64's arrow keys and space bar to control the onscreen players.) This training mode also will let you experiment with dif-

Tie Break is easy to learn. In no time, you'll have a racket in your hand, be ready to compete in a world tournament, or be ready to play against the computer in the training mode. The level of difficulty depends on such factors as the number of players you must beat in a tourna-



This player on a clay court has just hit a high lob to his singles opponent in Tie Break, DigiTek's new tennis game.

ferent rackets and surfaces. Being able to practice with other players is a nice feature of this game, since some of the other computerized tennis games let you practice only by hitting balls tossed by a ball machine.

No matter what mode you select, you don't have to worry about moving your player to the appropriate court position to hit the ball. Tie Break automatically positions the player for return shots. You simply decide how you're going to complete your return shot. Some of the options are volley, lob, slice, topspin, and smash. The computer will decide if your shot will be forehand or backhand. You control your shot's speed, direction, type, and length. It's also possible to pause during the action or slow the game down.

ment, the tension and weight of your racket, the opponent you select, and the court surface. This game's challenging to play, but it would be even more challenging if there were an option to position your player yourself for return shots as opposed to having the computer do it for you.

The manual provides you with enough information to get you started quickly. All types of shots are explained in enough detail so that you'll soon learn how to perform each one. There are even diagrams to show you in more detail exactly how to hit certain shots. The manual also includes useful information on how to use trick shots to defeat an opponent.

Tie Break's graphics and sound effects are great on the 64. You watch the

games as though you were looking down from above. You see the ball, players, court, and net in a three-dimensional view. You can hear the sound each time someone hits the ball or the ball bounces off the court. Tie Break lets you know visually when the ball hits the net or is out of bounds. You can see just what type of shots the players are taking, and the scores appear on the screen after each point is scored.

Overall, Tie Break is fun and challenging, and it'll give you hours of entertainment. So if tennis is your racket, I think you'll love playing this game, especially if the weather is too nasty to go outside on the courts. Bring the true-to-life action and excitement of tennis to your 64 and give this game your best shot. See you at Wimbledon!

CHRIS SAUCIER

Commodore 64 and 128—\$29.95

DIGITEK SOFTWARE
1916 Twisting Ln.
Wesley Chapel, FL 33543
(813) 973-7733

Circle Reader Service Number 341

CALC II

Years ago, I typed in a spreadsheet called SpeedCalc that I saw in Gazette. Since it has always loaded quickly, has been easy to use, and has provided correct answers with its calculations, I couldn't see any reason for switching to another spreadsheet. That was before I tried Calc II from Pankhurst Programming.

I'll have to admit that, at first glance, I wasn't impressed. I mean—a spreadsheet is a spreadsheet. They're made up of letters along the left side of the screen that identify rows and numbers run-

PUMP UP YOUR PRODUCTIVITY!

*Harness the potential of your
64 and 128 with these
powerful programs.*

Get more work out of your 64 and 128 with these two new disk products from COMPUTE's Gazette – the 1992 Best of Gazette Utilities, and the Gazette Graphics Grab Bag!

The 1992 Best of Gazette Utilities

Seize control of your operating system and your world!

Here's what's on it—MetaBASIC 64, MetaBASIC 128, Quick, Sprint II, Ultrafont+, RAMDisk 64, RAMDisk 128, BASSEM, SciCalc 64, List Formatter, MegaSqueeze.

The Gazette Graphics Grab Bag

Do it all with Commodore graphics!

Here's what's on it—Starburst Graphics, Screen Designer 128, 128 Graphics Compactor, 64 Animator, VDC Graphics, Dissolve 128, Super Slideshow, 128 Animator, 1526 PrintScreen, Supratechnic, Medium-Resolution Graphics, Screen Maker, GAS!64—Special Edition, GAS!128—Special Edition.



**ORDER
THEM
TODAY!**

Extend Your Computer Power With This Powerful Software!

YES!

I want to pump up my productivity! Please send me the disks checked below at \$11.95 each.

☐ The 1992 Best of Gazette Utilities

☐ The Gazette Graphics Grab Bag

☐ Subtotal

☐ Sales Tax (Residents of NC and NY please add appropriate sales tax for your area. Canadian orders, add 7% goods and services tax.)

☐ Shipping and Handling (\$2.00 U.S. and Canada, \$3.00 surface mail, \$5.00 airmail per disk.)

☐ Total Enclosed

☐ Check or Money Order ☐ MasterCard ☐ VISA

Credit Card No. _____ Exp. Date _____

Signature _____
(Required)

Daytime Telephone No. _____

Name _____

Address _____

City _____

State/Province _____ ZIP/Postal Code _____

MasterCard and VISA accepted on orders with subtotal over \$20.

Mail this coupon to COMPUTE's 1991 Utilities, 324 West Wendover Ave., Ste. 200, Greensboro, NC 27408.

ning across the top that identify columns. The intersection where a row and column meet is called a cell, and Calc II has plenty of those. In fact, it has 240 rows and 240 columns. A 64 doesn't have enough memory to use that many cells, but it does have room for about 20K of data. The actual number of free bytes remaining is displayed at the top of the Calc II screen.

I'm going to assume that everyone knows what a spreadsheet is and jump right into Calc II's strong points and improvements over its earlier version. One feature that's almost worth the price alone is Calc II's ability to split the screen into two, three, or four windows. This lets you see one part of the sheet while working in another section.

Many times I've been entering data in a budget with dates running across the top of the page and with income and expense categories running down the left. Things are fine as long as all the information is on one screen. By the time you've entered enough data to scroll across and down the page, however, you can no longer see the dates and the categories. After a while, you have to scroll back to the beginning to make sure you're entering data in the correct cells.

With Calc II you don't have this problem. You can open a couple of windows, lock the categories and dates in place, and then move anywhere on the sheet and still be sure you're not entering November's electric bill in the cell reserved for December's rent payment. You can divide the screen into any size windows you like.

Printing can also be a problem if your spreadsheet is larger than your printer can

handle. Calc II automatically splits at the end of a page, printing the rightmost sections on later pages. Other spreadsheets may wrap and print on the next line, destroying the row and column order. You can print just a portion of the sheet if you don't need all of it. You can also use embedded printer codes in Calc II to utilize your printer's compressed print modes.

Once you have data entered, you can manipulate it with Calc II's sorting routine. You can sort in ascending or descending order by row or column. This is the only area where I experienced any difficulty. I entered a group of names running down the page in a column. When I defined this block of data and asked the program to sort the column in ascending order, nothing happened. After several unsuccessful attempts, I asked it to sort the row, and then it worked. The same held true with data across the page. I call that a row, but the sort routine apparently considers that a column. Numbers sort with no problem, but text sorts according to word length unless cells are left-justified.

Data in cells can be justified left, right, or center, depending on your preference. Columns can be adjusted globally or individually to fit data of any width, from 3 to 38 characters. Other options include number of decimal places, auto or manual calculate, tape or disk, device number for printing, and screen and text colors. You can make these changes manually after a spreadsheet loads, but Calc II provides a separate program that lets you customize the program to fit your preferences. Then, anytime you run Calc II, it'll default to your cus-

tomized choices.

Another program that comes with Calc II is a merger program. Use it to combine different spreadsheet templates onto one larger spreadsheet. The resulting file can be a merger of any number of previously saved templates. Text and formulas must use different cells, however, or the second spreadsheet will overwrite those cells already in memory.

Speaking of templates, Calc II provides about 40 of them that are ready to load and run. Some of them help you calculate break-even points, business budgets, depreciation, loans, mortgages, standard deviations, and future worth of investments. Others will help balance your checkbook, keep a grade book, write sales slips, and figure factorials. One will even take room measurements and calculate how much wallpaper, paint, or carpeting you'll need for the job.

Several of the templates have borders and lines to set off data. In the past, I had to be content with a series of equal signs or dashes to create such effects. With Calc II, however, it's possible to use many of the Commodore keyboard graphics to create design elements that make your spreadsheet easier to read and more appealing to the eye.

Calc II treats mathematical calculations in the same fashion as Commodore BASIC, so you shouldn't have any problems entering your own formulas. It even accepts the Boolean operators And, Or, and Not.

Calc II also has a number of powerful built-in functions that can make your calculating chores easier. In addition to the usual Sums, Square Roots, Tangents, and Loga-

rithms—functions found in most spreadsheets—Calc II offers Rnd for random numbers, Peek to return a value of a memory byte, and Exp for exponents. Additional functions include Average, Minimum and Maximum values, and Number of Items in a block of data. Lookup and Index are functions used to find data in tables. Fix truncates remainders in calculations, and Round rounds the value up or down. An If function lets you make a choice between two options. How to use these and other functions is carefully explained in the manual.

A spreadsheet is a very powerful yet flexible tool that lets you perform a wide range of mathematical calculations. It's a productivity package you'll use again and again. If you don't have a spreadsheet, I recommend that you get one. Once you decide to get one, I heartily recommend Calc II.

TOM NETSEL

Commodore 64 and 128—\$29.00, plus \$4.95 shipping and handling

PANKHURST PROGRAMMING
P.O. Box 49135
Montreal, PQ
Canada H1N 3T6

Circle Reader Service Number 342

BAD BLOOD

Mutants get no respect! In most computer games, mutant creatures are monsters or enemies. Found roaming in dungeons or forests, these creatures are simply a source of experience points and useful objects left behind after termination. Bad Blood, from Origin, is an adventure game that takes place in a postapocalyptic world in which the mutants are the good guys and the humans are corrupt and vindictive. There's a lot of bad

SOFTWARE CLOSEOUTS

For Commodore 64 & 128

STRATEGIC SIMULATIONS INC. (SSI) \$19.50 EACH
 Typhoon of Steel, War of the Lance, Curse of
 Azure Bonds, Pool of Radiance, Countdown to
 Doomsday, Battles of Napoleon, Tony LaRussa
 Baseball, B-24, Roadwar 2000, Hillsfar, First
 over Germany, Heroes of Lance, Pro-Tour Golf,
 Battle for Normandy, Dragons of Flame or
 Dragonstrike.

MICROPROSE & MICROPLAY \$14.50 EACH
 Acrojet, Destroyer Escort, Netherworld, Rick
 Dangerous, Top Gunner, Xenophobe, Crusade in
 Europe, Decision in the Desert, Spitfire Ace,
 Hellcat Ace, Solo Flight, Silent Service, Conflict
 in Vietnam, Stunt Track Racer, 3D Pool, F-15
 Strike Eagle, Airborne Ranger, or Pirates!

ACCOLADE BARGAINS \$5.00 EACH
 PSI 5 Trading Co., Law of the West, Comics,
 Killed Until Dead, Hardball, Dambusters, Ace of
 Aces, Fight Night, or Jet Boys.

TAITO PREMIUM TITLES \$14.50 EACH
 Alcon, Arkanoïd, Arkanoïd II, Bubble Bobble, Qix,
 Operation Wolf, Rambo III, Rastan or Renegade.

ACTIVISION BARGAINS \$5.00 EACH
 Toy Bizarre, Mindshadow, H.E.R.O., Crossbow
 Championship Baseball, Zenji or Powerdrift.

ACTIVISION PREMIUM TITLES \$14.50 EACH
 Gee Bee Air Rally, Aliens, Ghostbusters II,
 Maniac Mansion, Beyond Dark Castle, Titanic,
 Space Shuttle, Christmas Model Kit, Mondu's
 Fight Palace, or Die Hard.

MINDSCAPE EDUCATION \$9.50 EACH
 MasterType/Writer Bundle, Perfect Score SAT,
 Shinobi, Gauntlet 2, Bop'n Rumble, Color Me,
 Combat Course, Into the Eagle's Nest, Indoor
 Sports, Boston Computer Diet, Songwriter,
 Thunderblade, Golden Oldies.

COMPSULT
 P.O. BOX 5160
 SAN LUIS OBISPO
 CA 93403-5160

AVALON HILL GAME COMPANY \$10.00 EACH
 Under Fire, MacBeth, Wooden Ships & Iron Men,
 Ripper!, Tsushima, Stocks & Bonds, Dr. Ruth,
 Parthian Kings, Jupiter Mission 1999, T.A.C.,
 Legionnaire, Tournament Golf, Maxwell Manor,
 Statis-Pro Baseball, Gulf Strike, Darkhorn,
 Guderian, Mission on Thunderhead, Panzers East,
 Super Sunday, or Panzer Jagd.

INFOCOM BARGAINS \$5.00 EACH
 Starcross, Suspended, Zork 2 or 3, or Deadline.

MORE GREAT DEALS!!!!
 Printed Wordge The Tool, by ValueWare \$2.75
 Partner 128 (cartridge for C/128 only),
 by Timeworks \$17.50
 Ghostwriter 128, by Hesware (C/128 only) \$14.50
 Super Expander 64 (cartridge), by Commodore \$5.00
 Assembler, by Commodore \$5.00
 Pet Emulator, by Commodore \$9.50
 Entertainer or Educator, by ValueWare \$3.75
 Tri-Math or Turtle Toyland, by HesWare \$4.50
 Attack of the Mutant Camels (cartridge), by HES \$3.50

Electronic Zoo \$16.50 each
 Iron Lord, Puffy's Saga, Skate Wars, B.A.T., Pick N' Pile
 Chessmaster 2100, by Software Toolworks \$14.50
 Dragon Wars or Neuromancer, Interplay \$19.50
 Heatwave or Steel Thunder, Accolade \$16.50
**TONS OF ADDITIONAL ITEMS FOR
 COMMODORE 64/128...IN STOCK NOW!
 CALL (805) 544-6616 TO CHECK TITLES**

VISA/MC ORDERS CALL TOLL-FREE
1-800-676-6616
 Credit Card Orders Only!!!! (\$25 minimum)

TO ORDER: Send check or money order, including
 shipping charges of \$5 for U.S.A., \$8 for Canada, \$12 all
 others. California addresses must include 7.25% sales
 tax. To receive our complete catalog of over 2,000 items
 for all computer types, send \$2 in cash or postage
 stamps. The catalog is FREE with any order. To check for
 an item not listed here, call (805) 544-6616.

**WE ALSO CARRY LOTS OF
 SOFTWARE FOR IBM, APPLE, MAC,
 AMIGA, ATARI & MORE!**

Circle Reader Service Number 181

**FLORIDA'S OLDEST & LARGEST
 COMMODORE/AMIGA
 FACTORY DIRECT DEALER
 AUTHORIZED SALES & SERVICE**

1-800-749-2225

**AMIGA-64/128-XT/286/386/486
 SOFTWARE-HARDWARE-SERVICE**

BRAND NEW!

AMIGA 500DS

512K RAM, 880K FLOPPY DRIVE, MOUSE, 4096 COLORS,
 WORD PROCESSOR, PAINT PROGRAM, FIA-18 INTERCEPTOR,
 F40 PURSUIT, INDIANA JONES & JOYSTICK

\$389.00

GENUINE COMMODORE 501 RAM EXP.....\$69.95
 GENUINE COMMODORE A1011 EXTERNAL FLOPPY DRIVES.....\$129.95
CALL FOR OTHER SUPER PRICES
NO TRADES NECESSARY!!! OFFER EXPIRES JUNE 30, 1992

NEW 1084S MONITORS.....\$299.00
AMIGA 520 VIDEO ADAPTERS.....\$39.95

CALL FOR FREIGHT CHARGES, OVERNIGHT, 2ND DAY & COD AVAILABLE. MASTERCARD, VISA, & DISCOVER
 ACCEPTED (VERIFICATION REQUIRED). FLORIDA RESIDENTS MUST ADD APPLICABLE SALES TAX. PURCHASE
 ORDERS ACCEPTED WITH PRIOR APPROVAL. ALL PRICES ARE IN US DOLLARS AND MUST BE PAID IN US FUNDS.
 SOFTWARE: NO REFUND! EXCHANGE FOR DEFECTIVES ONLY! HARDWARE: ALL RETURNS MUST HAVE AN RMA
 NUMBER. NON-DEFECTIVE RETURNS ARE SUBJECT TO A 20% RESTOCKING FEE. NO RETURNS AFTER 30 DAYS.

NEW AGE ELECTRONICS
13553 66TH STREET NORTH
LARGO, FL 34641

PH: (813) 530-4561 FAX: (813) 530-0799 BBS: (813) 360-7062

**COME VISIT US WHEN YOU VISIT FLORIDA
 ONLY 90 MINUTES FROM DISNEYWORLD IN
 SUNNY TAMPA BAY!**

Circle Reader Service Number 194

SAVE TIME AND MONEY

Yes, save time and money! Subscribe to the *Gazette Disk* and get all the exciting, fun-filled *Gazette* programs for your Commodore 64 or 128—already on disk!

Subscribe today, and month after month you'll get all the latest, most challenging, and fascinating programs published in the corresponding issue of *COMPUTE*.

New on the *Gazette Disk*! In addition to the programs that appear in the magazine, you'll also get outstanding bonus programs. These programs, which are often too large to offer as type-ins, are available only on disk—they appear nowhere else.

As another *Gazette Disk* extra, check out

"Gazette Gallery," where each month we present the very best in original 64 and 128 artwork.

So don't waste another moment. Subscribe today to *COMPUTE's Gazette Disk* and get 12 issues for only \$49.95. You save almost 60% off the single-issue price. Clip or photocopy and mail completed coupon today.

Individual issues of the disk are available for \$9.95 (plus \$2.00 shipping and handling) by writing to *COMPUTE*, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408.

YES! Start my one-year subscription to *COMPUTE's Gazette Disk* right away for only \$49.95.*

☐ Payment enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa

Acct. No. _____ Exp. Date _____

Signature _____ (Required)

Name _____

Address _____

City _____

State/Province _____ ZIP/Postal Code _____

Mail to *COMPUTE's Gazette Disk*, P.O. Box 3250, Harlan, IA 51593-2430

*Residents of NC and NY, please add appropriate sales tax for your area. Canadian orders, add 7% goods and services tax.



1991 Gazette Index

Everything's included!

Features, games, reviews, education/home applications, programming, bugswatter, feedback, and columns!

A superb interface includes pull-down menus, help screens, and keyboard, joystick, or mouse control. Features include super-fast searching and sorting capabilities. An options screen allows you to choose text colors, drive number, and input device. And there's full documentation on disk.

Choose from three modes of operation—*browse* for quick scanning, *view* for detailed information and descriptions, and *edit* for adding items from upcoming issues—and print to any printer. There's even a turbo-load option for maximum disk-access speed.

**ORDER YOUR
1991 GAZETTE
INDEX TODAY!**

(MasterCard and Visa accepted on orders with subtotal over \$20.)

To order, send \$7.95 per disk, the quantity of disks ordered, check or money order,* your name and complete street address:

**1991 Gazette Index
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408**

* Please add \$2 shipping & handling (\$5 foreign) for each disk (residents of NC, NJ, NY please add applicable sales tax; Canadian orders, add 7% goods and services tax).

All payments must be in U.S. funds. Please allow 4 weeks for delivery.

REVIEWS

blood here. Your goal is to bring peace to the Plains and prevent war between the mutant races and the humans who want to control and enslave them.

You start *Bad Blood* in the mutant town of Mardok. After watching an introduction, you choose a character for the game. The choices are Varigg, Jakka, and Dekker. Varigg is a green male, full mutant. Jakka is a female mutant, almost human except for her eye blasts. Dekker is a male human. All the characters have qualities that make them attractive alter egos. Their weaknesses make the game a different challenge with each character.

Bad Blood takes place on the Plains, a land laid to waste long ago by a nuclear bomb. The inhabitants managed to survive, but not without mutations. The humans tried to ignore the mutants and enslaved them. As time wore on, the mutants rebelled and formed their own communities. In addition to mountains and open spaces, the Plains consists of mutant towns Mardok, Okkarn, Nivvik, and Kitrum; human cities Xantium and Yvrium; and a dangerous shell of a city called Zero City.

Bad Blood's game screen features an overhead view of your character's surroundings in a mock television cabinet. Below this window on the left is a list of five commands; a jar of brown fluid on the right marks the character's life level. The commands are Inventory, Examine, Use, Talk, and Options (Pause game, Save game, Load game, and Music on/off). To play *Bad Blood*, move the character around in the top window with joystick or keyboard and select commands with the joystick button or space bar. The first three commands bring up an inventory window for further choices.

It's necessary to carry on a lot of conversations with people you meet in *Bad Blood*. They'll provide clues and other necessary information. Selecting the Use command brings up another menu with options to chat, ask a character about a specific topic from another list, or say goodbye.

Finally, in combat, you use a weapon or bare hands to attack by pressing the joystick button or space bar. That's all there is to *Bad Blood*. You'll be playing the game in no time.

Bad Blood possesses the right difficulty for beginners, and seasoned adventurers will enjoy the atmosphere of the game and the strange characters. The well-written manual details the history of the Plains and also explains the slang of the region, such as *tuff* (warrior or soldier), *hume* (human), and *mute*

(mutant). It's a fresh perspective to see the entire game on one scale, as is not the case in a lot of other role-playing games. You only see the interiors of buildings when you enter the doors. *Bad Blood* takes advantage of its game world and contains a lot of interesting mutants, not all of whom are friendly. Just remember to watch out for the reptilian Kejek creatures; they capture other mutants as slaves for the humans in exchange for big rewards.

The graphics in *Bad Blood* are top-notch. The overhead view is particularly impressive because of the detail it provides. Scrolling is smooth and fast. The only time the game halts and accesses the disk is when you enter a building or move into a new area of the Plains. Fortunately, this access time is short. *Bad Blood's* graphics bring the mutants to life and offer a rich environment for exploration. The catchy music is constant throughout the game and adds a lot to the enjoyment. Sound effects are not as prevalent.

Bad Blood is a fine example of how much more enjoyable a game can be with a simple control system. *Bad Blood's* interesting land and inhabitants make this game an ideal choice for adventure fans.

RUSS CECCOLA

Commodore 64 and 128—\$49.95

ORIGIN
P.O. Box 161750
Austin, TX 78716
(512) 328-0282

Circle Reader Service Number 343

**This publication is
available in microform
from UMI.**

Please send me information about the titles
I've listed below: _____

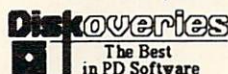
Name _____
Title _____
Company/Institution _____
Address _____
City/State/Zip _____
Phone (_____) _____

U·M·I

A Bell & Howell Company
300 North Zeeb Road, Ann Arbor, MI 48106 USA
800-521-0600 toll-free
313-761-4700 collect from Alaska and Michigan
800-343-5299 toll-free from Canada

LOTS OF DISKS! THE BEST in 64/128 PD.
GEOS, Graphics, Clipart, Demos, Games, Bible,
SID/MIDI, Educational, Basic 8
NEW! Graphics Scanning Service

Send stamp for FREE catalog or \$2 for sample disk.



Diskoveries
 PO Box 9153, Waukegan, IL 60079

Circle Reader Service Number 190

COMMODORE 64 PUBLIC DOMAIN

Highest Quality Since 1987*

Games, Education, Business, Utilities, GEOS, Music, Graphics & More. As low as 90¢ per collection. 1 stamp for complete catalog or \$2.00 for catalog AND 30 sample programs (refundable). 24 hour shipping.

64 DISK CONNECTION

4291 Holland Rd., Suite 562 • Virginia Beach, VA 23452
 (* Formerly RVH Publications)

Circle Reader Service Number 254

KODEKRAKR LTD.

Proudly presents a new innovation in software security check DE-protection! **PASSCODE KRAKR!** removes time consuming and often annoying documentation security checks in many of your games software programs. Lists are continually updated with the latest and greatest software releases available on disk! System also uses parameters which are easily updated with our "newest releases" list. Also available—**THE PARAMETER HOTLINE**—call on us to customize your favorite (registered owners only)! Don't let lost documents ruin another expensive program. TO ORDER CUSTOMIZATION PACKAGE, send \$21.95+\$4.00 S/H to:

KODEKRAKR LTD.
 761 Meade Lane • Virginia Beach, VA 23455

Circle Reader Service Number 223

Upgrade your Commodore system

Refurbished Hardware

MONITORS DRIVES OTHER
 1701 - \$235 1541 - \$100 1660 - \$30
 1702 - \$255 1541-II - \$120 1670 - \$50
 1801 - \$265 1571 - \$165 C64 - \$100
 1802 - \$285 1571-II - \$185 C64C - \$120
 1901 - \$295 1581 - \$180 C128 - \$175
 1902 - \$305 1001SFD - \$150 C128D - \$225
 1084e - \$325 1530 DATASETTE - \$35
 MANY BOOKS - \$10 SOFTWARE - \$10-20
 ASK FOR ANYTHING, I MIGHT HAVE IT!
J.P. PBM PRODUCTS BY MAIL
 P.O. BOX #1233, STATION B
 WESTON, ONTARIO, M9L2R9

New APROTEK modems

64/128/AMIGA-2400 BAUD - \$119
 64/128/AMIGA-1200 BAUD - \$89
 APROSAND-4 SLOT CARTRIDGE
 EXPANDER FOR THE 64/128 - \$40
New CMD accessories
 JIFFYDOS 64/128 & ANY DRIVE "SYSTEM" - \$85
 128D/ANY DRIVE "SYSTEM" - \$95
 ADDITIONAL DRIVE ROMS - \$45
 RAMLINK/RAMCARD C/W BATTERY (0Mb) - \$345
 1Mb RAMSIMP - \$75 4Mb RAMSIMP - \$250
 SHIPPING INCLUDED FOR CANADA, USA - \$10
 15 DAY WARRANTY ON REFURBISHED GOODS
 SEND A SASE FOR FREE INFO

C64/128 PUBLIC DOMAIN SOFTWARE

REQUEST FREE CATALOG or send \$2 for sample disk and catalog (REFUNDABLE). Categories include education, utilities, games, business, PRINT SHOP graphics, pre-tested programs and more. Rent for 75¢ or buy as low as \$1.00 per disk slide or for 80¢ for 70 or more. \$20 order gets 4 free disks of your choice.

NEXT DAY SHIPPING!

SINCE 1986



CALOE INDUSTRIES (P&G)
 PO BOX 18477, RAYTOWN, MO 64133



Circle Reader Service Number 255

COMPUTER REPAIR



205-739-0040

AUTHORIZED COMMODORE SERVICE CENTER

MOTHERBOARD REPAIR

64, 1541\$39.00

(Send Board Only)

128, 1571\$59.00

COMMODORE DISK DRIVES

1541\$49.00

MSD, 1571\$85.00

COMMODORE MONITORS

1701, 1802, 1084,\$85.00

1702, 1902\$85.00

COMMODORE COMPUTERS

C-64\$49.00

C-128\$85.00

SX-64 Portable\$79.00

COMMODORE PRINTERS

.....\$75.00

Amiga

Call for price

!!! FAST TURNAROUND !!!

All parts and labor included. \$7.50 shipping.
 \$15.00 APO and business equipment.

WE BUY ALIVE

C64 \$ CALL

1541 NEWT CALL

1541 ALPS CALL

SX64 CALL

1571, C128 CALL

1526/802 CALL

MUST BE COMPLETE

\$CALL ON DEAD

EQUIPMENT

CABLES AND MISC.

REG 6 PIN DIN \$ 4.45

12' 6 PIN DIN 6.95

18' 6 PIN DIN 7.95

MONITOR CABLES 4.45

TV SWITCHES 2.50

COMMODORE

KEYBOARD CALL

DSDD-Disk 25 for 11.95

PARTS

901 ROMS \$11.95

STR54041 12.50

41464 9.95

6560 PULLS 10.00

74LS629 4.95

82S100 12.95

325302 9.95

325572 9.50

8701 6.85

8502 9.00

6526 12.50

6522 4.50

6520 3.85

6532 6.29

6502 2.85

6510 9.95

6581 12.50

6567 19.95

4164 2.75

8721 16.85

C128 UPGRADE

ROMS 23.65

COMPUTER SHOPPE OF ALABAMA

1310 2nd Ave. NW, Suite C
 Cullman, AL 35055

DEALERS SEND FOR
 PARTS CATALOG

PRICE SUBJECT TO
 CHANGE

\$25.00 MIN ORDER

Circle Reader Service Number 178

Calc II

The Fastest
 Spreadsheet
 for the
 Commodore 64

Calc II makes your math work a breeze - whether it's a mortgage calculation, budgeting, or keeping sports statistics. Or use it for your non-math chores like organizing phone numbers or making a shopping list!

- Uses Commodore math routines for power and speed • gets results twice as fast as competitor's • Ideal for databases - sort by row or column • View lots of information fast with up to four configurable windows, row and column locking • Quick, responsive cursing • Independently adjustable decimal places, width and positioning

- of data • Uppercase, lowercase and Commodore graphics all available • Bar graphs on-screen with text • Over two dozen functions, including LOOKUP, AVG, IF, RND, SIN & FIX • 240 rows by 240 columns • Easy to remember commands • Uses disk or tape • Simple worksheet setup • Easy text entry. The Package includes a Detailed User's Guide with quick start info and spreadsheet tips. Also on disk with Calc II, two utility programs and over 40 ready to use worksheets. Only \$29.95 (plus \$4.95 shipping and handling). Please allow 4-7 weeks for delivery.

PANKHURST PROGRAMMING P.O. Box 49135 • Montreal • Quebec • Canada • H1N 3T6

Circle Reader Service Number 152

New for the C128! KeyDOS ROM!

The KeyDOS ROM is a chip for the empty socket inside your C128 and adds more than 40 new instantly available features only a keypress or two away!

20 KeyDOS F keys for simple "point & click" multiple drive access. All major DOS functions. Fast-load C64 programs in 128 mode, view SEQ files, execute batch files. Print, copy, view, scratch or rename multiple files. ASCII/CBM converter. Full 1581 subdirectory support! RAMDOS supports REUs up to 2MB. GEOS RBoot. Disk editor, ML debugger, alarm clock, screen dump editor.

LOTS MORE! Satisfaction Guaranteed! Write for more information.

Only \$32.50. See why C128 users say that KeyDOS ROM is a MUST!

Antigrav Toolkit, PO Box 1074, Cambridge, MA 02142

Shipping outside of US, Canada and Mexico add \$3

Circle Reader Service Number 155

BEGINNER BASIC

Larry Cotton

ALPHABET SPRITES

Let's transfer data from the 64's built-in character set to an interesting sprite format.

Lines 10-110 contain familiar sprite code, except for constant definitions in lines 50 and 60, which we'll look at in a moment. Line 120 begins a FOR-NEXT loop to copy three groups of character data.

Lines 130 and 170 access

```
RJ 10 PRINTCHR$(147):POK
    E646,7{10 SPACES}
JD 20 PRINTTAB(10)"PLEAS
    E WAIT 3 SEC.
RC 30 V=53248:REM 1ST SP
    R CTRL REGISTER
HD 40 POKEV+21,0:REM TUR
    N SPR 1 OFF
HG 50 X$="ABCDEFGHIJKLMN
    OPQRSTUVWXYZ":REM
    {SPACE}SEE LINE 33
    0
RA 60 S=6656:E=8:REM SEE
    LINE 140
BD 70 POKE2040,192:REM P
    OINT TO DATA
PC 80 REM POKEV+23,1:POK
    EV+29,1:REM ENLARG
    E
EP 90 POKEV+39,7:REM YEL
    LOW SPRITE
PX 100 X=160:Y=130:POKEV
    ,X:POKEV+1,Y:REM
    {SPACE}LOCATE
HM 110 FORB=12288TO12350
    :POKEB,0:NEXT:B=1
    2288:REM CLEAR SP
    RITE DATA AREA
FG 120 FORG=1TO3:GOSUB32
    0
SM 130 POKE56334,0:POKE1
    ,51:REM KBD OFF
DR 140 FORI=0TO21STEP3:Z
    =(T+S)*E:POKEB+I,
    PEEK(Z+A):REM COP
    Y CHARACTER DATA
XK 150 A=A+1:NEXT:A=0:B=
    B+1
GA 160 NEXT:B=B+21:GOTO1
    20
SB 170 POKE1,55:POKE5633
    4,1:REM KBD ON
AM 180 PRINTCHR$(147)
FB 190 POKE214,9:PRINT:P
    OKE211,17:PRINT"
    {3 SPACES}SS":REM
    3 SPACES BEFORE
    {SPACE}SS
MQ 200 POKE214,10:PRINT:
    POKE211,17:PRINT"
    {3 SPACES}SOR":RE
    M 3 SPACES BEFORE
    SOR
XB 210 POKE214,11:PRINT:
```

```
POKE211,17:PRINT"
KEYS"
XB 220 POKEV+21,1:REM SP
    RITE ON
AA 230 KB=197:SH=653:NK=
    64:MX=255
CG 240 K=PEEK(KB)+PEEK(S
    H):IFK=NKTHEN240
MB 250 IFK=7THENY=Y+1
SE 260 IFK=8THENY=Y-1
SC 270 IFK=2THENX=X+1
CB 280 IFK=3THENX=X-1
EQ 290 X1=INT(X/MX):X2=X
    -X1*MX
QF 300 POKEV,X2:POKEV+16
    ,X1:POKEV+1,Y:REM
    MOVE SPRITE
KG 310 GOTO240
JE 320 READA$:IFA$="*"TH
    EN170
HP 330 FORT=1TO26:IFA$=M
    IDS(X$,T,1)THENRE
    TURN
FG 340 NEXT
HD 350 DATA P,R,E,C,U,R,
    *
```

the character ROM located from 53256 through 53463. POKE 56334,0 turns off the keyboard, while POKE 1,51 switches the character ROM in. When the copying is complete, the two memory registers must be restored to normal with POKE 1,55 and POKE 56334,1. The order of poking in line 170 must be reversed from that of line 130.

Between lines 130 and 170, the ROM-to-sprite data copying occurs. We set up the sprite's shape data in memory registers 12288 through 12350; B is 12288. Recall that each sprite shape is defined by 63 bytes (three columns of 21 bytes). The first letter whose 8 bytes of data we'll transfer will appear in the upper left corner of the sprite, controlled by addresses 12288, 12291, 12294, and so on.

A FOR-NEXT loop from 0 to 21 in line 140 begins copying the eight bytes of character data. STEP 3 ensures that the data bytes fall under each other in the sprite so the character will be readable. Z is the ROM location of the character data. Its value is calculated by adding constant S and variable T and then multiplying by con-

stant E. (S and E are defined in line 60.) T comes from a subroutine in lines 320-350, which is called from line 120. Let's look at that subroutine.

A\$ is a letter of the alphabet. X\$ (the alphabet, defined in line 50) is scanned for a match. T indicates A\$'s place in the alphabet. Thus, if A\$ is the letter C, T will equal 3. The subroutine returns control to line 130. T is then added to S in line 140, which determines Z's value—the place to begin peeking for character data.

In line 140, the first time through the loop, B is poked with whatever's in Z. However, line 150 increments A (which starts as 0) by 1, so the next time through the loop, B+3 is poked with whatever's in Z+1. Thus, as the sprite data registers are increased by three, the character data registers are increased by one.

After the loop is finished in line 150, A is reset to 0, and B is increased by 1. The next two characters will be placed to the right of the first one in the sprite; the FOR-NEXT loop is finished in line 160. Finally, B is increased by 21, and another loop begins in line 120. This places three more characters under the first three.

Lines 180-310 print a message and illustrate a sprite-moving technique via the cursor keys. Line 230 looks at memory locations 197 and 653, which track keyboard presses. The sum of their peeked values equals 7, 8, 2, or 3, corresponding to down, up, right, and left cursor movements. Line 290 converts the sprite's horizontal position to pokable values for the two horizontal-position memory registers 53248 and 53264. Line 300 does the actual poking.

Experiment by deleting the first REM in line 80 and the printing in lines 190-210. Try other values for S in line 60, X\$ in line 50, and 26 in line 330. □

Here's an interesting way to turn keyboard characters into sprites, and move them about with cursor keys.

The GRAPEVINE GROUP Inc.™ COMMODORE UPGRADES

NEW POWER SUPPLIES

- A super-heavy, repairable C-64 power supply with an output of 4.3 amps (that's over 3x as powerful as the original). Featuring 1 year warranty, ext. fuse, schematics, UL approved. Cost is **\$37.95** and includes as a bonus the Commodore Diagnostic II (valued @ **\$6.95**).
- 4.3 amp supply for C-128. Same features as above—**\$39.95** (includes bonus package)
- **Our Biggest Seller** • 1.8 amp repairable heavy duty supply for C-64. (Over 120,000 sold) ... **\$24.95**

EMERGENCY STARTUP KITS

Repair your own Commodore/Amiga and save lots of money. Kits contain all major chips, schematics, diagnostics, etc. No soldering. Send for full details. Five different kits available.

512K RAM EXPANDERS

Super 1750 REU CLONE (512K). Does not require a larger power supply **\$142.50**
Original REU-1750 512K Expander Unit **\$131.00**

COMMODORE DIAGNOSTIC II

Originally developed as a software package, then converted to a readable format, the Diagnostician has become a fantastic seller. With over 38,000 sold worldwide, Diagnostician II utilizes sophisticated cross-reference grids to locate faulty components (ICs) on all C-64 and C1541 computers (C-128/64 mode). Save money and downtime by promptly locating what chip(s) have failed. (No equipment of any kind needed.) Success rate from diagnosis-to-repair is 98%. Includes basic schematic. **\$6.95** (Aval. for Amiga computers with 3 1/2" disk at **\$14.95**.)



We Ship Worldwide

Send For Free 36 Page Catalog

3 CHESTNUT ST., SUFFERN, NY 10901

Order Line 1-800-292-7445 Fax 914-357-6243

Hours: 9-6 E.S.T. M-F 914-357-2424 Prices subject to change

Tell a friend you've heard it through the Grapevine.

Circle Reader Service Number 145

SPECIALS

- **COMPUTER SAVER:** This C-64 Protection System saves you costly repairs. Over 52% of C-64 failures are caused by malfunctioning power supplies that destroy your computer. Installs in seconds between power supply & C-64. No soldering. 2 year warranty. An absolute must and great seller **\$17.95**
- **PRINTER PORT ADAPTER** by Omnitronix. Avoid obsolescence. Allows you to use any Commodore (C-64) printer on any PC compatible or clone. Does not work with Amiga **\$34.95**

PRINthead REFURBISHING

Save time and money by having your tired, worn-out or damaged printhead refurbished or remanufactured at a fraction of the cost of a new one. Features low cost, 5 day service and 1 year warranty. For example: Okidata 80/90/100 heads are \$64.95; Epson (9 Pin) EX/FX/LX are \$69.95.

REPLACEMENT/UPGRADE CHIPS & PARTS

6510 CPU	
6526 CIA	
6581 SID	
6567 Video	
PLA 906114	
All 901/225-6-7-9	
4164 (C-64/RAM)	.60
C-128 ROMs Upgrade (set 3)	24.95
C1571 ROM Upgrade (310654-05)	10.95
C-64 Keyboard (new)	19.95
Commodore Cables	Call
Service Manuals for C64, C128, 1802, 1084, 1541	\$21.95

\$9.95
NEW LOW PRICES
EACH



Big Blue Reader 128/64 - 4.0

Transfers word processing, text, ASCII, and binary files between C64/128 and IBM PC compatible 360K 5.25" and 720K 3.5" disks.

New Version 4.0 features: Transfers ASCII, PET ASCII and Screen Code files including: WordWriter, PocketWriter, SpeedScript, PaperClip, WriteStuff, GEOS, EasyScript, Fleet System and most others.

Supports drives # 8-30. New Backup (C128) and Format (1571/1581) programs. Reads MS-DOS sub-directories, uses joystick, and more.

Includes C64 & C128 programs. **Requires 1571 or 1581 Disk Drive.**

Big Blue Reader 128/64 - 4.0 only \$44.95

Version 4.0 upgrade, send original BBR disk plus \$18.

Bible Search 3.2

1. Entire Old and New Testament text on 4-1541/71 or 2-1581 disks.
2. Exhaustive English Concordance on 2-1541/71 or 1-1581 disks; includes more than 700,000+ references.
3. Incredible five (5) second look-up time per word, per/disk.
4. Instant, automatic spell checking of more than 12,800 words.
5. Boolean search options, including AND, OR & NOT logic.
6. Search the entire Bible in 5 seconds with 1581 or HD (v3.52).
7. Money Back Guaranteed!

Includes: C64 & C128 programs; printer and disk output; users guide, disk case. Available on (7) 1541/71, or (4) 1581 disks.

KJV \$49.95 ☆☆☆☆ **NIV \$59.95**

Any questions? Call or write for more information.

Also available! Amiga, Bible Search

Order by check, money order, or COD. US funds only.

FREE shipping in US. No Credit Card orders.

Canada & Mexico add \$4 S/H, Overseas add \$10 S/H (\$5 BBR)
SOGWAP Software (219)724-3900

115 Bellmont Road; Decatur, Indiana 46733



8 BIT
PO BOX 542
LINDENHURST, NY
11757-0542
(516)-957-1110

6 PAC # B: 6 DISK SET FOR ONLY \$5.00

MARIO BROS. - Super Mario Bros, Mario II, Marjos Brewery, ARKANOID - Arkonoid II, Krakout Pro2 & Pro4, Adictaball
GENEBOLOGY - (2 disks) PEDIGREE V3.0 Your family tree!
TRIVIA CONST. SET - Make your own Trivia Quiz Game
COMPUTER AIDED DESIGN - C.A.D. V3.0

6 PAC # C: 6 DISK MUSIC SET FOR \$5.00

T.V. THEMES - Hill St., MacGyver, Letterman, L.A. Law, More
MOVIES - (w/picture) Somewhere Out There, NY, NY, & More
ASSORTMENT - Danger Zone, Elvis Tribute (w/picture), More
60's POP - Light My Fire, California Dreamin', Fire & Rain, More
PROGRAMS - Pitch Pipe, Piano, Organ, ET Theme, Bach, More
SOUND F/X - Tarzan, U2, Warp, Vulcan Mind, Try/Think, More

ADD \$2.00 SHIPPING FOR TOTAL ORDER
U.S. FUNDS ONLY! SORRY NO C.O.D.'S
A COPY OF OUR CURRENT CATALOG
WILL BE SENT WITH ALL ORDERS!
OR CALL TO RECEIVE A COPY FREE
MONDAY - FRIDAY 10:00AM TO 5:00PM EST

Circle Reader Service Number 162

DISKS O'PLENTY INC

7958 PINES BLVD. SUITE 270A
PEMBROKE PINES FL 33024
(305) 963-7750

Call or write for free descriptive catalog of
C64/128 Public Domain & Shareware
Choose from over 900 Disks
Adult list of over 50 Disks available
to those 18 or over.

Choose 6 for \$5.00
LIMITED OFFER

- | | |
|-------|---------------------|
| 021MU | SID MUSIC UTILITIES |
| 019GR | PRINTSHOP UTILITIES |
| 019ED | JR HIGH EDUCATION |
| 062ED | HIGH SCHOOL EDUC. |
| 033ED | TYPING / SPANISH |
| 031ED | COMPUTER SCIENCE |
| 010UT | PIRATES TOOLBOX |
| 119GA | FOREIGN ARCADE |
| 022GA | CASINO-BOARD GAMES |
| 021GE | GEOS FONTS |
| 002MS | LOTTERY PROGRAMS |
| 003MS | COLLECTORS CORNER |

Circle Reader Service Number 253

Can Your
Computer Make
YOU

\$1,000,000?

WITH LOTTERY PC YOUR NEXT TICKET
COULD BE WORTH MILLIONS!

LOTTERY uses the raw power and storage of your computer to determine and refine the number selection methods that will win the various lottery games you play. Don't be limited to the one or two methods that other programs use, they might not work in your state. There is no better system available!

Join the growing list of winners using our system.

SPECIFY:

Lottery 64(C64/128)

Lottery PC

IBM PC/XT/AT and compatibles

Commodore 64/128 & Plus/4 are registered trademarks of Commodore Int.

IBM PC/XT/AT are registered trademarks of International Business Machines Inc.

To order, send \$29.95 for each plus \$3.00 postage & handling per order to:
(Illinois residents add 6% sales tax)
(Orders outside North America add \$3.00)



C.O.D. orders call:
(708) 566-4647



Superior Micro Systems, Inc.

26151 N. Oak Ave.
Mundelein, IL 60060



Circle Reader Service Number 221

Give.

American Heart
Association



MACHINE LANGUAGE

Jim Butterfield

ROTATING BUFFERS

You may recall an earlier expedition into hexadecimal number printing. In this column, we'll expand on that subject.

Here's what this program will do. The user names a program file. Our program examines that file and reports two things: the load address and whether or not the last three bytes of the file are 0. The load address of a program gives you a hint as to whether this is a normal program (generated with a SAVE command) or an unusual one that will need to be brought in with a LOAD "filename",8,1 command. Normal load addresses vary from computer to computer, but these addresses almost always end with hexadecimal digits 01.

If the load address ends with digits 01, our program will read the rest of the file to catch the last three bytes. If each of the three are 0, the program will print ALL BASIC.

For example, a program with a load address of \$0801 that ends with three binary 0s is more than likely a BASIC language program that was probably saved on a 64. That means it can be loaded with the normal ,8 extension.

The program must read all the bytes of the file in order to catch the last three, but it doesn't need to save all those bytes that it encounters first. A buffer large enough to hold three bytes is all that's required. As the data bytes come from the file, they'll be stored in one of these three addresses. A pointer will move across the three locations, looping around as needed. In other words, we have a rotary buffer. When we finally see the END OF FILE signal, the contents of those three locations will be the pieces of informa-

tion that we want to know.

The BASIC program first pokes the machine language code in place. Then it asks for a filename and opens logical file 1. The machine language program is called in to do the main job. When control returns to BASIC, the file is closed, and the program is finished.

Note that the file is opened and closed from BASIC. Although we could do the same job from machine language, we'd save little time or memory. It's nice to have BASIC on hand for such jobs. On other computers, BASIC wouldn't be there, and you'd lose this easy and flexible option. Since it's built into the Commodores, why not use it?

The machine language program is situated at addresses \$2000 to \$2081. When called, it connects to logical file 1 and reads the first two bytes into addresses \$2200 and \$2201. Those bytes are converted to hexadecimal and printed.

At address \$2021 we check to see that the last two digits of the load address are 01. To do this, simply examine the contents of \$2200. If you don't find value 1 there, skip to the end of the program.

Our rotary pointer is held in X; the value is temporarily stored in location \$2202 when we read in a new byte from the file. The data byte is stored in one of three locations in the range \$2203 to \$2205.

```
2027 8E 02 22 STX $2202
202A 20 E4 FF JSR $FFE4
202D AE 02 22 LDX $2202
2030 9D 03 22 STA $2203,X
2033 E8      INX
2034 E0 03   CPX #$03
2036 D0 02   BNE $203A
2038 A2 00   LDX #$00
203A A5 90   LDA $90
203C F0 E9   BEQ $2027
```

Note that ST, the status byte, is at \$90 for Commodore computers from the VIC-20 on.

For early PET/CBM machines, substitute address \$96.

Once the file is completely read, it's easy to examine the three bytes of the buffer to see if they're all 0.

If the file does end with three 0 bytes, the program prints its ALL BASIC message. The loop to do this is at locations \$204B to \$2057.

The machine language program disconnects from the file by calling Kernal subroutine \$FFCC, prints a Return, and then passes control back to BASIC.

Full coding isn't given here, but you can disassemble it to see the details. Note the changes for older PETs on lines 110 and 270.

```
10 DATA 162,1,32,198,255,32,
20 DATA 228,255,141,0,34,32
30 DATA 228,255,141,1,34,169
40 DATA 36,32,210,255,162,1
50 DATA 189,0,34,32,96,32
60 DATA 202,16,247,174,0,34
70 DATA 202,208,49,142,2,34,
80 DATA 32,228,255,174,2,34
90 DATA 157,3,34,232,224,3
100 DATA 208,2,162,0,165
110 DATA 144:REM 150 FOR PET
120 DATA 240,233,169,0,162,2
130 DATA 29,3,34,202,16,250
140 DATA 170,208,13,162,0
150 DATA 189,119,32,32,210
160 DATA 255,232,224,11,208
170 DATA 245,32,204,255
180 DATA 169,13,76,210,255
190 DATA 72,74,74,74,74,32
200 DATA 107,32,104,41,15
210 DATA 120,248,24,105,144
220 DATA 105,64,216,88,76,210
230 DATA 255,32,65,76,76,32
240 DATA 66,65,83,73,67,13
250 FOR J=8192 TO 8321:READ X
260 POKE J,X:T=T+X:NEXT J
270 IF T<>14512 THEN STOP:
    REM 14518 FOR PET
300 INPUT "NAME OF PROGRAM FILE";F$
310 OPEN 15,8,15:OPEN 1,8,3,F$
330 INPUT#15,E,E$,E1,E2
340 IF E<>0 THEN PRINT
    E;E$;E1;E2:END
350 SYS 8192
360 CLOSE 1
```

Here's a method to determine a program's starting address and whether it's a BASIC or a machine language program.

WORLD VIEW

Emil Heyrovsky

VIEW FROM CZECHOSLOVAKIA

Don't worry if you hear us talking favorably about Commies in Czechoslovakia these days. The Communist party hasn't ruled in our country since November 1989. When we talk about Commies now, we're talking about our Commodore computers.

Czechoslovakia is a free country in the heart of Europe; its population slightly exceeds 15 million. It's a mountainous country with a colorful history.

Now back to the Commies. Thanks to the Communist regime, the value of the Czechoslovak crown has fallen drastically in comparison to its pre-World War II value. The present exchange rate is about 30 crowns to the U.S. dollar.

Now hold your breath. The average monthly salary of a Czechoslovak citizen today is 2,700 crowns. That's about \$90. During the Communist rule, a 64 was available only through special shops for almost 10,000 crowns (\$330). A 1541 disk drive unit cost about the same amount.

At that time, though, the average salary was much lower, and the exchange rate was absurd. The only rational way to get a computer was to import it from Germany or another West European country. Thanks to bureaucratic Communist formalities, this was never an easy process. Nowadays, a 64 costs about 5000 crowns (\$165). Since it's now easy to travel throughout Europe and the formalities are minimal, most people prefer to buy their 64s in Germany.

Compared with other 8-bit-ers, 64s are probably the most widespread machines here; Sinclair Spectrums formerly held that distinction. There are also some Atari, Sord, and Sharp computers

around. As for 16-bit machines, the Amiga 500 is very popular, as is the Atari line. IBM compatibles are purchased by business people, but Apples and Macs are seldom seen.

Commodore software isn't available in stores here. As far as I know, there has never been any means of buying software for our beloved computers. As a result, 90 percent of all programs around here are pirated.

Here's another reason why pirating is so widespread: A standard game costs about 750 crowns (\$25), which is about one-third of an average monthly salary. Only a madman would consider buying software at that price. All over Europe there are teams of software crackers providing copies of pirated software, and programs without copy protection spread quickly.

The most common way to get software on disk or tape, which is still popular here, is to swap programs with your friends. Another method is to pick up a bunch of disks or a couple of tapes and head for a Commodore club. There are two of them here in Prague, and trading is popular. Members might swap Bard's Tale 3 for Last Ninja 3 or Boulder Dash for Maniac Mansion, or possibly buy a 1541 brochure. One club publishes instruction manuals for programs and peripherals as well as for computers. The club is combined with an Amiga group, which has a bad effect on weaker individuals, who, instead of squeezing bytes out of their 64s, bounce off to Amigaland.

Club meetings usually take place twice a month in a rented hall. Some tables are put together to form a counter for selling books and brochures; other tables are used for trading and copying. Copying is performed on 64s or 128s with

disk drive units or Datasette tape recorders, which often have some kind of homemade adapter for smooth copying. Members or visitors bring equipment from home, since the club doesn't provide computers. Once a month, lectures on computer topics are held. Most of the people you meet at a club are young.

I've noticed recently that the number of Commodore owners is slowly decreasing. I'll try to explain why. The 64 and 128 are generally considered to be affordable game machines. A person who wants to do word processing or some other "serious" application thinks of buying an IBM or compatible. Amigas are bought by wealthier game players (or serious users). Because some people have never learned to use their Commies fully, they don't think much of them, and they want to get rid of them.

Modems are almost unknown here. This is due partly to our rotting telecommunication system and also to public ignorance of electronic bulletin boards and E-mail. Moreover, long-distance calls are extremely expensive. Just a normal chat with an overseas friend can swallow and digest your salary in a jiffy.

Many new computer magazines are showing up on our newsstands, but several of them are just translations of German or other imported publications. Unfortunately, we don't get COMPUTE, and there isn't a good magazine available here that's dedicated to the 8-bit Commodores. That's too bad because, as I said, many Commodore owners in Czechoslovakia often never really learn what great machines they have.

Emil Heyrovsky lives in Prague. He is the coauthor of Padlock (January 1992). □

The 64 and 128 remain popular in Czechoslovakia, along with tape drives, user groups—and software pirates.

Steve Vander Ark

GRAPHIC SCRAPS

Since desktop publishing and writing are what I do most with GEOS, I collect graphics, along with utilities to handle them as efficiently as possible. In earlier columns I've mentioned some of the better sources for GEOS graphics—Lamb Art & Design, for example—and suggested ways to convert graphic images from other formats into GEOS.

After you've collected all these terrific graphics, you still have to get them into your documents. Back in the August 1991 column I mentioned Scrap It, a utility which clips photo scraps out of geoPaint documents. Since many collections of artwork for GEOS come in that form, Scrap It is essential. Other collections come in prepared photo albums, such as the clip art files from Susan Lamb (3575 East County 18th Street, Yuma, Arizona 85365) and those from GeoWorks itself (GEOS Clip Art Disk, 2150 Shattuck Avenue, Berkeley, California 94704). Graphics have to make it into photo scrap form, if they're going to be of any use, because photo scraps are the way GEOS moves images between applications.

A photo scrap is simply a small chunk of bitmap, the dot pattern that makes up a graphic image. The GEOS operating system tags photo scraps as system files and handles them in a special way. You can't rename a photo scrap, for example, since applications are designed to look for the filename PHOTO SCRAP when you select a tool to import a graphic into your document. There can be only one such file with that filename on any given disk at a time. Photo albums are collections of such scraps, and the key to using photo scraps efficiently is to handle albums efficiently.

The primary tool for handling photo albums is the photo manager desk accessory (currently in version 2.1, which works in 40- and 80-column modes). Since the photo manager can access any of the photo albums on the disk, you use it to find the image you want and copy or cut it into an individual scrap. That scrap can then be imported using the Edit menu functions.

So far that's basic GEOS operation, and you've likely mastered it already. But the more graphics you collect, the more you wish for a way to access and organize them better. The photo manager performs its job admirably, but it can't access scraps on another drive. This means your photo albums, which are tremendous disk-space eaters, must be specially prepared for each project, with only the images you'll require, or copied in and out as needed. But then if you want to create an album of only the clips you're likely to need, there's no easy way to move them from one album to another without opening and closing each album to make a transfer.

Now if you've read this column before, you know that I only pose dilemmas if I have a solution in mind. In the case of photo scraps, there are several excellent utilities available which can make all this shuffling of graphics a breeze. The program AlbumCopy by Michael Myers presents you with a control panel which allows you to select source and destination albums and then flip through the clips to choose which ones you'd like to transfer. It can't get much easier than that! To download it from Q-Link, request ALBUMCOPY, which was uploaded by GeoLib PH.

This utility is also handy for reducing the size of any over-size photo albums you've creat-

ed with earlier versions of GEOS that the GEOS 2.0 photo manager can't handle. This happens because the version 2.0 albums hold only half as many clips as those of the older model—60 as compared to 120. If you'd rather just revert to the old version so you can work with the old photo manager, get a copy of Album Reverter, written by Joe Buckley (ALBREV.SFX on Q-Link, uploaded by Red Storm). This utility reverses the update process that the new photo manager automatically performs when you try to open an older-style album.

But the single most helpful program you can get for using photo scraps comes in two incarnations, one a desk accessory and the other an application. Scrapgrab, the desk accessory version, lets you select a photo from an album or a scrap on a disk in a different drive. The application version, identical except that it handles larger scraps, is called Photograb. To download from Q-Link, look for SCRAPGRAB and PHOTOGRAB. GeoLib PH uploaded both of these.

With these utilities you can finally keep your photo collection where it belongs—on a graphics collection disk—and access your clip art when you need it from within geoPublish or other applications without those albums taking up every inch of disk space on your main work disk. The addition of these two files will let you effortlessly use your clip art, not be strapped down by it.

Imagine a disk of photo albums, all sorted into categories, which you can access, browse through, and select from at will as you geoPublish along. You'll never again have to drop out of your document to track down a graphic when you keep your graphics on a library disk and Scrapgrab on your work disk. □

**The more
graphics you collect,
the more
you wish for a way
to access
and organize them
better.**

The Gazette Productivity Manager

(Formerly PowerPak)

Harness the productivity power of your 64 or 128!

Turn your Commodore into a powerful workhorse, keep track of finances, generate reports in a snap, manage your money in minutes—all with the new 1991 Gazette Productivity Manager! Look at all your 64/128 Productivity Manager disk contains.

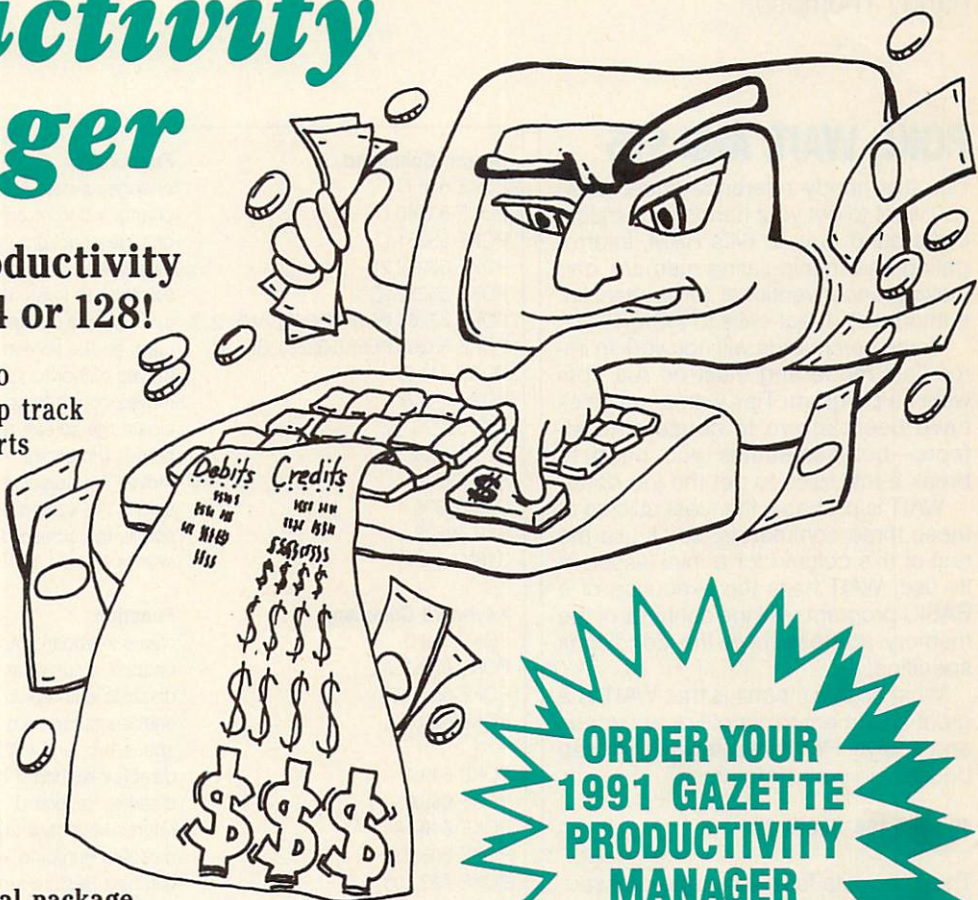
GemCalc 64 & 128—

A complete, powerful, user-friendly spreadsheet with all the features you'd expect in an expensive commercial package (separate 64 and 128 versions are included). Most commands can be performed with a *single* keypress!

Memo Card—Unleashes the power of a full-blown database without the fuss! **Nothing's easier**—it's a truly simple computerized address file. Just type in your data on any one of the index cards. Need to edit? Just use the standard Commodore editing keys. Finished? Just save the data to floppy. What could be easier?

Financial Planner—Answers all of those questions concerning interest, investments, and money management that financial analysts charge big bucks for! You can plan for your children's education and know exactly how much it will cost and how much you need to save every month to reach your goal. Or, decide whether to buy or lease a new car. Use the compound interest and savings function to arrive at accurate estimates of how your money will work for you. **Compute the answer at the click of a key!**

DON'T MISS OUT ON THIS POWERFUL WORKHORSE!



**ORDER YOUR
1991 GAZETTE
PRODUCTIVITY
MANAGER
TODAY!**

(MasterCard and Visa accepted on orders with subtotal over \$20).

☐ **YES!** Please send me ___ Productivity Manager disk(s) (\$14.95 each).

___ Subtotal

___ Sales Tax (Residents of NC and NY please add appropriate sales tax for your area. Canadian orders, add 7% goods and services tax.)

___ Shipping and Handling (\$2.00 U.S. and Canada, \$3.00 surface mail, \$5.00 airmail per disk.)

___ Total Enclosed

___ Check or Money Order ___ MasterCard ___ VISA

Credit Card No. _____

Signature _____ (Required)

Daytime Telephone No. _____

Name _____

Address _____

City _____

State/Province _____ ZIP/Postal Code _____

Send your order to Gazette 1991 Productivity Manager,
324 W. Wendover Ave., Ste. 200, Greensboro, NC 27408.

PROGRAMMER'S PAGE

Randy Thompson

POKE, WAIT, AND SYS

Use this handy reference sheet when you want to get your hands dirty mucking around in your 64's RAM, interrogating and manipulating memory, and making unconventional (and often unauthorized) direct calls to ROM.

Some commands will not work in immediate mode and must be run from within a program. Tips with an asterisk have been known to cause side effects—but sometimes you need to break a few rules to get the job done.

WAIT is probably the least utilized of these three commands, so I'll use the rest of this column for a mini lesson in its use. WAIT halts the execution of a BASIC program until the contents of the memory address meet the conditions specified.

What all this means is that WAIT is a great statement to simplify key fetches and delays. Here are a few ways it can be used in your programs.

10 WAIT 198,15:GET AS

This line waits for a key to be pressed.

10 WAIT 197,63:KEY=PEEK(197)

20 WAIT 197,64

Line 10 waits for a key to be pressed, and line 20 waits for it to be released.

10 WAIT 653,1

20 WAIT 643,1,1

Line 10 waits for Shift to be pressed, and line 20 waits for it to be released. Change the 1 in each line to 2, and the program waits for the press and release of the Commodore key. Substitute a 4 to wait for the Ctrl key.

Contributors to this list of coding voodoo are Michael Hall of De Soto, Texas; Jerry Krebs of Taylor, Texas; Stacy Olivas of Graham, Washington; and Helen Roth of Los Angeles, California.

"Programmer's Page" is interested in your programming tips and tricks. Send them to Programmer's Page, COMPUTE's Gazette, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408. We'll pay \$25-\$50 for each tip we publish. □

Screen Command

POKE 646,C
POKE 53280,C
POKE 53281,C
POKE 53272,21
POKE 53272,23
POKE 53265,PEEK(53265) AND 239
POKE 53265,PEEK(53265) OR 16
POKE 211,C
POKE 214,R
SYS 59296
SYS 58726
SYS 59516
SYS 59626
SYS 59848
SYS 59749

Function

changes cursor color (0-15)
changes border color (0-15)
changes background color (0-15)
switches to uppercase mode
switches to lowercase mode
turns off the screen display
turns on the screen display
moves cursor to specified column (0-79)
moves cursor to specified row (0-24)
clears the screen and prints READY
homes the cursor
moves the cursor down
scrolls the screen up
scrolls the screen down*
inserts a line

Keyboard Command

POKE 198,0
POKE 650,128
POKE 650,127
POKE 650,0

POKE 649,1
POKE 649,0
POKE 649,10
POKE 808,239
POKE 792,193
POKE 808,234
POKE 808,237:POKE 792,71
WAIT 653,1
WAIT 653,2
WAIT 653,4

Function

clears keyboard buffer
enables autorepeat for all keys
disables autorepeat for all keys
enables autorepeat for only the cursor keys, space bar, and Inst/Del key (computer's default)
disables keyboard buffer
disables keyboard
returns keyboard to normal
disables Run/Stop key*
disables Restore key
disables Run/Stop-Restore and LIST*
reenables Run/Stop-Restore and LIST
waits for Shift to be pressed
waits for Commodore key to be pressed
waits for Ctrl to be pressed

BASIC Command

POKE 818,32
POKE 818,237
POKE 775,200
POKE 775,167
POKE 774,0
POKE 774,24
POKE 204,0
POKE 204,255
POKE 19,65
POKE 19,0

Function

disables SAVE
enables SAVE
disables LIST
enables LIST
prevents line numbers from listing
resurrects lines numbers
turns on cursor during GET*
turns off cursor during GET
turns off question mark during INPUT*
turns on question mark during INPUT

System Command

SYS 58266
SYS 65126
SYS 64738

Function

prints startup message and erases BASIC program
warm reboot (erases screen)
cold reboot (erases program)

Miscellaneous Command

POKE 56325,R
POKE 54296,15:POKE 54296,0
WAIT 56320,16,16
WAIT 56321,16,16
SYS 62913

Function

changes cursor blink rate (0-255, 58 is normal)
makes a click sound
waits for fire button of joystick in port 2 to be pressed
waits for fire button of joystick in port 1 to be pressed
prints the name of the latest loaded or opened file

PROGRAMS

FILE LOGGER

By Roger Bachelder

File Logger is a short two-part program that will list all the files that you have on a disk. It presents you with the disk's BAM title, the disk ID, and each file's starting track and sector. File Logger's unique feature is that it also gives you each file's starting and ending address.

Typing It In

The first part of File Logger is written entirely in BASIC. To help avoid typing errors, use The Automatic Proofreader to enter the program. See "Typing Aids" elsewhere in this section. After entering the first part of the program, be sure to save it to disk before exiting.

The second part of File Logger is written in machine language. Use MLX, our machine language entry program, to type it in. Again, see "Typing Aids." When MLX prompts, respond with the following values.

Starting address: 6AA0

Ending address: 6E07

Save this machine language program as LOG.ML on the same disk as the BASIC program. File Logger will automatically load and run LOG.ML, so be sure to save it with this filename.

Reading Disks

After the program runs, you'll see a menu giving you the option to LOG DISK or EXIT TO BASIC. Insert the disk you want to check and enter 1. You'll then be asked whether or not you want the output sent to a printer.

File Logger will then read the directory and present you with the disk name, ID, file type, track, sector, and blocks. It will then ask you if you want to see the starting and ending addresses. It prints this information in hexadecimal notation.

FILE LOGGER

```
HE 10 IFPEEK(830)=1 THEN 720
CQ 15 IFPEEK(27296)<>169 THEN LO
AD"LOG.ML",8,1
QJ 17 POKE808,225:POKE146,0
XS 20 POKE 53281,6:POKE53280,6
:CLR
JR 25 DIMF$(100):TP$(0)="XXX":
TP$(2)="PRG":TP$(1)="SEQ"
```

```
HG 30 TP$(3)="USR":TP$(4)="REL"
"
QD 40 H$="0123456789ABCDEF"
RH 50 PRINTCHR$(142);"{CLR}
{DOWN}{BLK} (C) 1992 COM
PUTE PUBLICATIONS INTL L
TD"
RA 51 PRINT"{10 SPACES}ALL RIG
HTS RESERVED"
JE 55 PRINT"{DOWN}{WHT}
{9 SPACES}F I L E
{3 SPACES}L O G G E R"
DH 60 PRINT"{3 SPACES}{35 T}"
DQ 80 PRINT"{DOWN}{28 SPACES}"
AJ 90 PRINT"{3 DOWN} 1 --
{YEL}LOG DISK"
CD 100 PRINT"{DOWN}{WHT} 2 --
{SPACE}{YEL}EXIT TO BAS
IC"
PH 120 PRINT"{2 DOWN}{WHT}PLEA
SE SELECT (1 - 2):";:PO
KE204,0
FX 130 GETAS
QG 140 IFAS="1" THEN 180
KP 150 IFAS="2" THEN POKE204,1:P
RINTAS:END
KR 170 GOTO130
QK 180 PRINT"{CLR}{5 DOWN}
{WHT} SEND OUTPUT TO PR
INTER (Y/N):";:POKE204,
0
QE 190 GETAS
RE 200 IFAS="Y" THEN POKE204,1:P
RINTAS:Z=4:GOTO230
PQ 210 IFAS="N" THEN POKE204,1:P
RINTAS:Z=3:GOTO230
RD 220 GOTO190
MR 230 PRINT"{CLR}{4 DOWN}
{WHT}LOOKING AT DIRECTO
RY...";:BMS$="":OPEN15,8
,15,"I":OPEN3,8,3,"S"
DK 240 CLOSE4:OPEN4,Z:DN=Z-3:T
B=10*DN
CE 250 FORX=1 TO 142:GET#3,AS:NE
XT:PRINT".";
ES 260 FORX=1 TO 16:GET#3,B$:BMS$
=BMS$+B$:NEXT:PRINT".";
KX 270 GET#3,AS,AS:GET#3,11$,I
2$
EP 275 IFZ=4 THEN PRINT#4,CHR$(1
3);CHR$(13)
CC 280 PRINT"{CLR}";:PRINT#4,T
AB(TB);"{8 SPACES}BAM T
ITLE{13 SPACES}ID"
CA 290 IFZ=3 THEN PRINT"{WHT}
{8 SPACES}{24 T}"
GG 295 IFZ=4 THEN PRINT#4,TAB(TB
);"{8 SPACES}-----
"
RF 300 IDS=11$+12$:PRINT"{WHT}
";:PRINT#4,TAB(TB);"
{8 SPACES}";:PRINT#4,BM
$;"{6 SPACES}";IDS
FJ 305 IFZ=4 THEN PRINT#4,CHR$(1
3);CHR$(13)
GP 310 PRINT"{2 DOWN}{WHT}";:P
RINT#4,TAB(TB);"
{3 SPACES}TRK SEC BLKS
```

```
{4 SPACES}FILENAME
{5 SPACES}TYPE"
KA 320 IFZ=3 THEN PRINT"{WHT}
{3 SPACES}{33 T}"
SP 325 IFZ=4 THEN PRINT#4,TAB(TB
);"{3 SPACES}-----
"
BG 330 FORJ=1 TO 92:GET#3,AS:NEX
TJ
BS 340 M=M+1:GET#3,K$,T$,S$:IF
S$=" " THEN S$=CHR$(0)
PJ 350 F$="":FORJ=1 TO 16:GET#3,
AS:F$=F$+AS:NEXTJ
MR 360 FORJ=1 TO 10:GET#3,AS:NEX
TJ
JH 370 L=0:IFAS<>" " THEN L=ASC(A
S)
DX 380 PRINT#4,TAB(TB);"
{3 SPACES}";:PRINT"
{WHT}";
KR 390 L$=RIGHT$(STR$(L),2):IF
LEFT$(L$,1)=" " THEN L$="
"+RIGHT$(L$,1)
BP 400 IFASC(T$+CHR$(0))=0 THEN
IFL=0 THEN 560
KH 410 T$=RIGHT$(STR$(ASC(T$)),
2):IFLEFT$(T$,1)=" " TH
ENT$="0"+RIGHT$(T$,1)
GK 420 S$=RIGHT$(STR$(ASC(S$)),
2):IFLEFT$(S$,1)=" " TH
ENS$="0"+RIGHT$(S$,1)
AH 430 K=ASC(K$+CHR$(0))-128
EQ 440 IFK<10 OR K>5 THEN K=0
SP 450 PRINT#4,T$;"{2 SPACES}"
;:PRINT"{WHT}";
SP 460 PRINT#4,S$;"{2 SPACES}"
;:PRINT"{WHT}";
BS 470 PRINT#4,L$;"{3 SPACES}"
;:PRINT"{WHT}";
BH 480 K$=TP$(K):PRINT#4,F$;
{SPACE}";:PRINT"{WHT}";
DE 490 IFK=0 THEN PRINT"{WHT}";
EA 500 PRINT#4,K$:IFK>2 THEN 52
0
QJ 510 F$(Q)=F$:Q=Q+1
PM 520 GET#3,AS:IFM<8 THEN GET#3
,AS,AS:GOTO540
EQ 530 M=0
RC 540 IFST=0 THEN 340
CS 550 CLOSE3
ES 560 PRINT:PRINT"{DOWN}{YEL}
DISPLAY {WHT}START
{YEL}& {WHT}END {YEL}AD
DRESSES ({WHT}{YEL})/
{WHT}{N(YEL)} {WHT}";:PO
KE204,0
PQ 570 GETAS
MJ 580 IFAS="Y" THEN POKE204,1:P
RINTAS:GOTO610
PJ 590 IFAS="N" THEN POKE204,1:P
RINTAS:POKE830,0:GOSUB9
00:RUN
CP 600 GOTO570
JX 610 SYS27296
XP 620 PRINT#4,CHR$(13);CHR$(1
3):PRINT"{CLR}"
HA 630 CLOSE3:X=-1
```



```

AX 640 PRINT "{WHT}";:PRINT#4,T
      AB(TB);"{5 SPACES}FILEN
      AME{9 SPACES}START
      {3 SPACES}END"
CK 645 IFZ=4THENPRINT#4,TAB(TB
      );"{5 SPACES}-----
      "
CA 650 IFZ=3THENPRINT "{WHT}
      {5 SPACES}{29 T}"
FE 660 X=X+1:IFZ$(X)=" "THENPOK
      E830,0:GOSUB900:RUN
RC 670 PRINT "{WHT}";:PRINT#4,T
      AB(TB);"{5 SPACES}";F$(
      X);" ";
DC 680 FORV=16TO1STEP-1:P$=MID
      $(F$(X),V,1)
PC 690 IFP$=CHR$(160)THENNEXTV
KH 700 F$=LEFT$(F$(X),V)
JF 710 POKE830,1:LOADFS,8,1
GP 720 A=PEEK(828):B=PEEK(829)
AH 730 PRINT "{WHT}";:GOSUB780
XC 740 PRINT#4,"{2 SPACES}";
JH 750 A=PEEK(175):B=PEEK(174)
EF 760 PRINT "{WHT}";:GOSUB780
SB 770 PRINT#4:GOTO660
FP 780 PRINT#4,"$";:W=A:GOSUB7
      90:W=B:GOSUB790:RETURN
XP 790 J=INT(W/16):I=W-J*16
XA 800 A$=MID$(H$,J+1,1):B$=MI
      D$(H$,I+1,1)
XA 810 PRINT#4,A$;B$;:RETURN
PJ 900 PRINT:PRINT
PC 905 PRINT "{HOME}";{23 DOWN}
      {BLU}";{7 SPACES}PRESS AN
      Y KEY TO CONTINUE"
SB 910 FORDE=1TO100:GETA$:IFA$
      <>" "THEN950
KP 920 NEXT
ME 930 PRINT "{HOME}";{23 DOWN}
      {WHT}";{7 SPACES}PRESS AN
      Y KEY TO CONTINUE"
CH 940 FORDE=1TO100:GETA$:IFA$
      <>" "THEN950
FR 945 NEXT
GF 946 GOTO905
CD 950 RETURN

```

LOG.ML

```

6AA0:A9 C0 8D 30 03 A9 6A 8D 50
6AA8:31 03 60 00 00 A9 A5 8D 62
6AB0:30 03 A9 F4 8D 31 03 60 7A
6AB8:00 8C 3C 03 8E 3D 03 60 38
6AC0:85 93 A8 D0 13 8C EF 6D 77
6AC8:B1 BB C9 24 F0 0A A6 BA 98
6AD0:E0 08 90 04 E0 0C 90 05 C7
6AD8:A5 93 4C A5 F4 B1 BB 99 C8
6AE0:F0 6D C8 C9 3A D0 03 8C E6
6AE8:EF 6D C4 B7 90 EF 8C EE 71
6AF0:6D 20 AF F5 A5 11 8D 12 78
6AF8:6C AD 00 DD 8D 13 6C AD 8C
6B00:15 D0 8D 14 6C A9 59 A0 E5
6B08:6C 85 AE 84 AF A9 00 A0 59
6B10:03 8D 0F 6C 8C 10 6C A9 9B
6B18:57 20 E7 6B A0 00 B1 AE 6D
6B20:20 A8 FF C8 C0 20 90 F6 5C
6B28:20 AE FF 18 98 65 AE 85 79
6B30:AE 90 03 E6 AF 18 98 6D CD
6B38:0F 6C 8D 0F 6C 90 03 EE EE

```

```

6B40:10 6C C9 A7 AD 10 6C E9 5E
6B48:04 90 CC A9 02 A0 03 8D 9F
6B50:0F 6C 8C 10 6C A9 45 20 11
6B58:E7 6B 20 AE FF 78 AD 00 2A
6B60:DD 29 03 85 11 09 20 8D A3
6B68:00 DD 2C 00 DD 70 FB 18 FC
6B70:AD 11 D0 29 07 69 2F 8D D8
6B78:24 6C A9 00 8D 15 D0 85 99
6B80:AF 20 15 6C AC 00 6E 30 13
6B88:3C F0 05 A9 FF 8D 01 6E 9B
6B90:A2 01 A5 AF D0 17 AE 02 EA
6B98:6E AC 03 6E 20 B9 6A A5 7B
6BA0:B9 D0 04 A6 C3 A4 C4 86 34
6BA8:AE 84 AF A2 03 A0 00 E8 9B
6BB0:BD 00 6E B1 AE E6 AE D0 8E
6BB8:02 E6 AF EC 01 6E 90 EF E1
6BC0:AD 00 6E D0 BC 20 D4 6B C4
6BC8:98 30 06 A6 AE A4 AF 18 A2
6BD0:60 4C 30 F5 AD 12 6C 85 64
6BD8:11 AD 13 6C 8D 00 DD AD A2
6BE0:14 6C 8D 15 D0 58 60 8D 16
6BE8:0E 6C A5 BA 20 B1 FF 24 2E
6BF0:90 30 13 A9 FF 20 93 FF C0
6BF8:A0 00 B9 0C 6C 20 A8 FF 4D
6C00:C8 00 06 90 F5 60 20 D4 7D
6C08:6B 4C 07 F7 4D 2D 45 02 B5
6C10:03 20 00 97 00 2C 00 DD 7A
6C18:50 FB A0 00 38 A6 11 EA 95
6C20:AD 12 D0 E9 32 90 04 29 12
6C28:07 F0 F5 8E 00 DD 8A 09 FD
6C30:2D AA EA EA 24 80 AD 00 4E
6C38:DD 4A 4A EA 4D 00 DD 4A FA
6C40:4A EA EA EA 4D 00 DD 4A 75
6C48:4A 45 11 4D 00 DD 8E 00 23
6C50:DD EA 99 00 6E C8 D0 C4 03
6C58:60 F0 3F A9 08 8D 00 18 AE
6C60:A2 01 A5 22 D0 02 A9 12 95
6C68:85 06 86 07 A9 B0 20 25 3C
6C70:03 A9 E0 20 25 03 C9 02 1E
6C78:B0 F2 8D 00 18 60 A2 00 9F
6C80:86 F9 8D 98 02 9D 5B 02 95
6C88:4C C6 D5 85 06 86 07 18 BC
6C90:68 69 01 8D 76 03 68 8D 0F
6C98:77 03 A5 06 20 D1 03 A9 FA
6CA0:06 85 31 20 0A F5 50 FE CD
6CA8:B8 AD 01 1C 91 30 C8 D0 DA
6CB0:F5 A0 BA 50 FE B8 AD 01 40
6CB8:1C 99 00 01 C8 D0 F4 20 A9
6CC0:E0 F8 20 E9 F5 C5 3A F0 17
6CC8:05 A9 05 4C 69 F9 4C 1E DD
6CD0:04 0F 07 0D 05 0B 03 09 84
6CD8:01 0E 06 0C 04 0A 02 08 8B
6CE0:00 20 C5 03 A0 00 B9 00 23
6CE8:06 4A 4A 4A 4A 4A 4A 4A 2C
6CF0:06 29 0F AA BD 78 03 AA 24
6CF8:A9 01 8D 00 18 2C 00 18 22
6D00:D0 FB 8E 00 18 8A 0A 29 3C
6D08:0F 8D 00 18 68 AA BD 78 31
6D10:03 8D 00 18 0A 29 0F EA 4F
6D18:8D 00 18 C8 D0 C8 A9 08 4E
6D20:8D 00 18 4D 00 1C 8D 00 25
6D28:1C 60 38 85 06 E5 22 F0 85
6D30:41 49 FF 18 69 01 0A AE 91
6D38:00 1C A8 30 03 CA 88 2C B2
6D40:E8 C8 8A 29 03 85 44 AD 0A
6D48:00 1C 29 FC 05 44 8D 00 73
6D50:1C A9 8F 8D 05 18 2C 05 54
6D58:18 30 FB 98 D0 D9 A5 06 93
6D60:85 22 20 4B F2 85 43 AD 21
6D68:00 1C 29 9F 1D 1A 04 8D 50

```

```

6D70:00 1C 60 00 20 40 60 A9 CA
6D78:12 A2 01 20 32 03 30 5A 7F
6D80:A9 02 85 0A A8 B9 00 06 34
6D88:29 87 C9 82 D0 21 AE 96 3A
6D90:04 BD 97 04 D9 03 06 F0 E7
6D98:08 C9 2A F0 23 C9 3F D0 CD
6DA0:0E C8 E8 EC 95 04 90 E9 68
6DA8:A9 A0 D9 03 06 F0 11 18 1A
6DB0:A5 0A 69 20 90 CC AE 01 26
6DB8:06 AD 00 06 D0 BD F0 1A DB
6DC0:A4 0A C8 8C 50 01 D0 03 7D
6DC8:20 88 03 BE 01 06 B9 00 B5
6DD0:06 F0 11 30 0F 20 32 03 70
6DD8:10 EE 8C 50 01 A9 FF 8D 4A
6DE0:00 06 30 E4 A9 12 20 D1 39
6DE8:03 A9 01 4C 69 F9 08 00 D7
6DF0:31 39 39 31 20 4C 4F 47 05
6DF8:2D 4D 2F 4C A0 A0 A0 D1
6E00:F6 00 00 00 00 00 00 00 58

```

Roger Bachelder does his program-
ming in Salem, Oregon.

DEMO MAKER

By Danny English

Creating a colorful, eye-catching demon-
stration can be a difficult task, but it's a
snap with Demo Maker. You can create
and edit rasters, text, scrolling effects,
and many other options with the touch of
a key. When you save your finished de-
mo on disk, it can be loaded and run like
a BASIC program.

You can use Demo Maker as a loader
with instructions to load and execute an-
other program, or you can use it to make
an exciting title to your own creation. The
vivid raster display will add luster to the
dullest program.

Typing It In

Demo Maker consists of two programs.
The first part is written entirely in ma-
chine language. You must enter this pro-
gram with MLX, our machine language
entry program. See "Typing Aids" else-
where in this section. When MLX
prompts you, respond with the values
given below.

Starting address: 3F7B
Ending address: 4A7A

When you've finished typing, be sure
to save the program to disk with the
name DEMO.CODE. The second part
of Demo Maker will load this program
by that name.

Demo Maker's editor is written en-
tirely in BASIC. To help avoid typing er-

rors, use The Automatic Proofreader to enter this program. See "Typing Aids" again.

Raster Magic

When both programs are saved on disk, load the BASIC Demo Maker and type RUN. The demo code will be loaded, followed by the menu screen.

Creating a demo with the editor is very simple. Use the cursor keys to move the pointer up and down the options. To change colors, move the pointer to the color you wish to change and press the + or - keys. The colors will cycle forward or backward to the color you desire.

The raster colors are all coded by shades. For example, shades of red include brown, pink, and orange. Shades of yellow are actually a rainbow of colors. For effect, the large scrolling raster is split in half, allowing two shades to be seen at once.

Text Options

The demo screen has a scrolling text display in the lower screen, fading or flashing text in the middle, and large title text at the top. There are options on the menu to edit all of these. To select one of the options, move the pointer and press Return.

Because of the way the text is stored, a special text-entry routine is used that doesn't allow you to use the cursor keys. You simply type from left to right, however, and the cursor will wrap around normally. The Delete key does work in its normal fashion. Press Return when you've finished entering, and the text will be stored in the demo.

The large title letters were created using a special character set. This character set doesn't contain punctuation marks or numbers; therefore, use only letters A to Z when entering the title.

In the center of the screen is a large section of text that's fading or flashing. The menu has options to control the color and speed of the fading. Notice that the top and bottom lines of this text appear on rasters in the demo. The color of these lines is specified in the menu by the text on raster color.

Saving the Demo

You can preview your demo at any time by pressing the space bar. Press-

ing the space bar again will return to the menu. When the demo looks the way you want it, move the pointer to the save option and press Return. You'll then be asked if it should boot another file. If you enter Y, you'll be prompted for the filename and SYS address if it's a machine language program. If you enter N, the program will tell the demo to return to BASIC after viewing.

The next prompt is the filename for the actual demo. Saving the program takes more than a minute, but then it can be loaded and run like a BASIC program. If the demo is to boot another file, be sure to save it on the same disk. After watching the demo, press the space bar to continue.

Feel free to experiment with color combinations and creative text displays. Whether you're making the title to a game or just an attractive display, Demo Maker will amaze you.

DEMO.CODE *← should be 0B*

```
3F7B:00 0B 08 0A 00 9E 32 30 6D
3F83:36 34 00 00 00 49 00 8C DB
3F8B:EA EA EA EA EA EA EA EA
3F93:EA EA EA EA EA EA EA EA 12
3F9B:EA EA EA EA EA EA EA EA 1A
3FA3:18 A9 78 85 FB 69 85 85 16
3FAB:25 A9 0A 85 FC 69 08 85 E3
3FB3:26 18 A9 00 65 FB 85 27 CD
3FBB:A9 40 65 FC 85 28 A2 FF AD
3FC3:A0 00 8C FE CF 8C FF CF 94
3FCB:F0 04 B1 25 91 27 C6 25 28
3FD3:E4 25 D0 02 C6 26 C6 27 CB
3FDB:E4 27 D0 02 C6 28 C6 FB 31
3FE3:E4 FB D0 E6 C6 FC E4 FC 4D
3FEB:D0 E0 A5 28 C9 A0 B0 0A 7E
3FF3:85 34 85 38 A5 27 85 33 7E
3FFB:85 37 4C B2 46 20 20 20 D2
4003:20 20 20 20 20 20 20 83
400B:20 20 20 20 20 20 20 8B
4013:20 20 20 20 20 20 20 93
401B:20 20 20 20 20 20 20 9B
4023:20 20 20 20 20 20 05 01 4E
402B:13 19 20 14 0F 20 03 15 D4
4033:13 14 0F 0D 09 1A 05 20 CF
403B:09 0E 14 0F 20 19 0F 15 CF
4043:12 20 0F 17 0E 20 04 05 26
404B:0D 0F 21 21 20 20 20 20 2E
4053:20 20 20 20 20 20 20 D3
405B:20 20 20 20 20 20 20 DB
4063:20 20 20 20 20 20 20 E3
406B:20 20 20 20 20 20 20 EB
4073:20 20 20 20 20 20 2A FD
407B:20 12 01 13 14 05 12 20 DA
4083:03 0E 0C 0F 12 13 20 20 F8
408B:20 20 20 20 2A 20 14 05 29
4093:18 14 20 03 0F 0C 0F 12 32
409B:13 20 20 20 20 20 20 95
40A3:20 20 20 20 20 20 20 24
40AB:20 20 20 20 20 20 20 2C
```

```
40B3:20 20 20 20 20 20 20 34
40BB:20 20 20 20 20 20 20 3C
40C3:20 20 20 20 20 20 2A 4E
40CB:20 06 01 04 09 0E 07 20 EC
40D3:14 05 18 14 20 20 20 C5
40DB:20 20 20 2A 20 14 09 7D
40E3:14 0C 05 20 0C 05 14 14 C4
40EB:05 12 13 20 20 20 20 B9
40F3:20 20 20 20 20 20 20 74
40FB:20 20 20 20 20 20 20 7C
4103:20 20 20 20 20 20 20 85
410B:20 20 20 20 20 20 20 8D
4113:20 20 20 20 20 20 2A 9F
411B:20 14 05 18 14 20 13 03 1F
4123:12 0F 0C 0C 13 20 20 20 2E
412B:20 20 20 2A 20 01 0E AD
4133:09 0D 01 14 09 0F 0E 20 9F
413B:20 20 20 20 20 20 20 BD
4143:20 20 20 20 20 20 20 C5
414B:20 20 20 20 20 20 20 CD
4153:20 20 20 20 20 20 20 D5
415B:20 20 20 20 20 20 20 DD
4163:20 20 20 20 20 15 13 C2
416B:05 20 01 13 20 01 20 04 13
4173:09 13 10 0C 01 19 2C 20 E6
417B:0F 12 20 02 0F 0F 14 20 2B
4183:01 0E 0F 14 08 05 12 20 C5
418B:06 09 0C 05 20 20 20 07
4193:20 20 20 20 20 20 20 16
419B:20 20 20 20 20 20 20 1E
41A3:20 20 20 20 20 20 20 26
41AB:20 20 20 20 20 20 20 2E
41B3:20 20 20 20 20 20 20 36
41BB:20 20 20 20 20 20 20 3E
41C3:20 20 20 10 12 05 13 13 41
41CB:20 13 10 01 03 05 20 20 C1
41D3:20 20 20 20 20 20 20 56
41DB:20 20 20 20 20 20 14 08 2E
41E3:05 20 03 0F 0D 10 15 14 29
41EB:05 20 04 05 0D 0F 20 0D BB
41F3:01 0B 05 12 20 09 13 20 E6
41FB:0E 0F 17 20 02 0C 01 13 83
4203:14 09 0E 07 20 01 03 12 23
420B:0F 13 13 20 19 0F 15 12 81
4213:20 13 03 12 05 05 0E 2E 74
421B:2E 2E 20 19 0F 15 20 03 E7
4223:01 0E 20 03 15 13 14 0F 0C
422B:0D 09 1A 05 20 14 08 09 76
4233:13 20 04 05 0D 0F 20 09 08
423B:0E 14 0F 20 19 0F 15 12 F0
4243:20 0F 17 0E 20 17 09 14 E2
424B:08 20 14 08 05 20 13 09 B6
4253:0D 10 0C 05 20 0D 05 0E 81
425B:15 2D 04 12 09 16 05 0E 10
4263:20 05 04 09 14 0F 12 2E 79
426B:20 20 05 04 09 14 20 03 C4
4273:0F 0C 0F 12 13 2C 20 14 23
427B:05 18 14 2C 20 14 09 14 45
4283:0C 05 13 2C 20 06 01 04 93
428B:09 0E 07 2C 20 01 0E 04 E0
4293:20 0D 0F 12 05 21 20 20 7B
429B:14 08 05 20 04 01 1A 1A 41
42A3:0C 09 0E 07 20 12 01 13 01
42AB:14 05 12 20 04 09 13 10 3A
42B3:0C 01 19 20 03 01 0E 20 FB
42BB:02 05 20 15 13 05 04 20 AC
42C3:01 13 20 01 20 04 05 0D C9
42CB:0F 2C 20 0F 12 20 05 16 09
42D3:05 0E 20 01 20 02 0F 0F A8
42DB:14 20 14 0F 20 01 0E 0F 16
```


PROGRAMS

```

42E3:14 08 05 12 20 06 09 0C 6D 4513:F0 F0 F0 F8 00 7F 70 70 4F 4743:4C BC FE 8D 20 D0 8D 21 60
42EB:05 2E 20 20 14 08 05 20 6F 451B:70 70 7F FF 00 E0 E0 E0 10 474B:D0 60 A9 C8 8D 16 D0 A9 2C
42F3:10 0F 13 13 09 02 09 0C 46 4523:70 78 3F 0F 00 70 70 70 EF 4753:15 8D 18 D0 60 0A 0A 0A 29
42FB:09 14 09 05 13 20 01 12 A8 452B:70 70 7F FF 00 7F 78 78 61 475B:08 08 08 02 02 02 00 00 29
4303:05 20 05 0E 04 0C 05 13 03 4533:78 78 7F FF 00 7F 78 78 6F 4763:00 0B 0B 0B 0C 0C 0C 0F 7E
430B:13 21 20 20 15 0E 0C 05 67 453B:78 78 78 FC 00 E0 E0 E0 25 476B:0F 0F 01 01 01 01 01 01 84
4313:01 13 08 20 14 08 05 20 CC 4543:70 78 3F 0F 00 7F 78 78 64 4773:0F 0F 0F 0F 0C 0C 00 00 80
431B:07 12 01 10 08 09 03 20 55 454B:78 78 78 FC 00 03 03 03 24 477B:00 09 09 09 02 02 02 02 22
4323:10 0F 17 05 12 20 0F 06 DD 4553:03 03 03 1F 00 00 F8 70 D4 4783:08 08 0A 0A 0A 00 06 0E 64
432B:20 19 0F 15 12 20 36 34 EC 455B:70 78 3F 1F 00 7F 79 78 7F 478B:03 01 01 03 0E 06 00 0B BF
4333:2E 2E 2E 20 2A 2A 20 14 72 4563:78 78 78 FC 00 78 78 78 72 4793:0C 0F 01 01 0F 0C 0B 00 DA
433B:08 05 20 03 0F 0D 10 15 1D 456B:78 78 7F FF 00 70 70 70 53 479B:00 00 05 02 00 01 01 00 F0
4343:14 05 20 04 05 0D 0F 20 F3 4573:70 70 70 F8 00 70 70 70 03 47A3:00 05 20 20 20 57 45 4C AE
434B:0D 01 0B 05 12 20 2A 2A D9 457B:70 70 70 F8 00 E0 E0 E0 1E 47AB:43 4F 4D 45 20 54 4F 20 BE
4353:20 20 02 19 20 04 01 0E E4 4583:70 78 3F 0F 00 7F 70 70 8C 47B3:54 48 45 20 20 20 20 43 2E
435B:0E 19 20 05 0E 07 0C 09 31 458B:70 70 70 F8 00 E1 E1 E0 34 47BB:4F 4D 50 55 54 45 20 44 E0
4363:13 08 20 20 20 20 20 20 5D 4593:70 78 3F 0F 00 7F 71 70 9E 47C3:45 4D 4F 20 4D 41 4B 45 7F
436B:20 20 20 20 00 00 07 0F 2D 459B:70 70 70 F8 00 03 00 00 24 47CB:52 20 C2 C4 CE CD 47 AD 1A
4373:1F 3C 78 F0 F0 00 FF 7F BD 45A3:F8 78 3F 0F 00 03 03 03 B6 47D3:CD 47 C9 BF F0 01 60 A9 46
437B:70 70 70 70 7F 00 0F 3F C4 45AB:03 03 03 07 00 70 70 70 5C 47DB:C7 8D CD 47 A0 00 B9 71 C9
4383:78 70 E0 E0 E0 00 FF 7F 13 45B3:70 38 3F 0F 00 0E 0E 07 B8 47E3:07 99 70 07 C8 C0 27 D0 43
438B:70 70 70 70 70 00 FF 7F 7E 45BB:07 03 03 01 00 73 77 7F 37 47EB:F5 A0 00 B1 FE C9 00 D0 A8
4393:78 78 78 78 7F 00 FF 7F 86 45C3:7E 7C 78 7F 00 03 07 0E F2 47F3:09 A9 E0 85 FE A9 41 85 8C
439B:78 78 78 78 7F 00 0F 3F 6C 45CB:1C 38 70 F8 00 03 03 03 25 47FB:FF 60 8D 97 07 A5 FE 18 B2
43A3:78 70 E0 E0 E0 00 FC 78 26 45D3:03 03 03 07 00 07 0F 1E C9 4803:69 01 85 FE A5 FF 69 00 29
43AB:78 78 78 78 7F 00 1F 03 6B 45DB:3C 78 FF FF 00 1E FE FE 18 480B:85 FF 60 A2 00 A0 00 B9 D0
43B3:03 03 03 03 03 00 03 00 2B 45E3:1E 1E 1E 3E 00 FC 1E 0E EA 4813:A5 47 C9 20 D0 05 E8 E8 D8
43BB:00 00 00 00 00 00 00 78 B4 45EB:0E 3E FC F0 00 00 00 00 BB 481B:4C 38 48 38 E9 40 9D 50 48
43C3:78 78 78 79 7F 00 FC 78 B9 45F3:1C 3C F8 E0 00 0E 0E 0E 2B 4823:04 18 69 40 9D 51 04 18 3F
43CB:78 78 78 78 78 00 F0 78 61 45FB:0E 1C FC F0 00 F0 10 00 27 482B:69 40 9D 78 04 18 69 40 4F
43D3:7C 7E 77 73 71 00 F0 78 44 4603:00 02 FE FE 00 E0 20 00 A3 4833:9D 79 04 E8 E8 C8 C0 14 FF
43DB:7C 7E 77 73 71 00 0F 3F 4F 460B:00 00 00 00 00 7E 7E 0E 9C 483B:D0 D5 A2 00 00 00 B9 B9 30
43E3:78 70 E0 E0 E0 00 FF 7F 73 4613:1C 3C F8 E0 00 FC 3C 3C 92 4843:47 C9 20 D0 05 E8 E8 4C E4
43EB:70 70 70 70 7F 00 0F 3F 35 461B:3C 3C 3C 7E 00 C0 C0 C0 89 484B:67 48 38 E9 40 9D A0 04 05
43F3:78 70 E0 E0 E0 00 FF 7F 83 4623:0C C0 C0 F8 00 78 78 33 4853:18 69 40 9D A1 04 18 69 E2
43FB:70 70 70 70 7F 00 0F 3F 45 462B:78 F0 F0 E0 00 80 C0 E0 C0 485B:40 9D C8 04 18 69 40 9D 51
4403:78 70 78 3E 0F 00 FF FF 4F 4633:70 38 1C 3E 00 00 00 00 6D 4863:C9 04 E8 E8 C8 C0 14 D0 C7
440B:83 03 03 03 03 00 F8 70 21 463B:00 08 F8 F8 00 1C 1C 1C 3D 486B:D5 60 A0 00 99 70 DB C8 22
4413:70 70 70 70 70 00 F8 70 EA 4643:1C 1C 1C 3E 00 FC 7C 3C 75 4873:C0 28 D0 F8 60 A0 00 99 37
441B:70 38 38 1C 1C 00 F8 70 F5 464B:1C 1C 1C 3E 00 0E 0E 0E B6 487B:50 D8 C8 C0 A0 D0 F8 60 2A
4423:70 70 70 70 71 00 F8 70 03 4653:1C 3C F8 E0 00 E0 00 00 AD 4883:A0 00 99 68 D9 99 F8 D9 1F
442B:38 1C 0E 07 03 00 F8 70 83 465B:00 00 00 00 00 0E 8E EE 2C 488B:C8 C0 FF D0 F5 60 A0 00 30
4433:38 1C 0E 07 03 00 FF FF 29 4663:7C 3C FC E6 00 E0 C0 E0 31 4893:99 40 D9 99 F8 DA C8 C0 5B
443B:80 00 00 01 03 00 C0 E0 8E 466B:70 38 1C 3E 00 F0 78 1C 76 489B:28 D0 FD 60 EE F0 CF AD C1
4443:F0 78 3C 1E 1E 00 E0 F8 77 4673:1C 3C F8 E0 00 80 80 80 CD 48A3:F0 CF CD 97 47 B0 17 AC 0C
444B:3E 0E 0E 1C F8 00 C0 F8 3C 467B:80 80 80 C0 00 1C 1C 1C 49 48AB:F1 CF B9 91 47 20 83 48 83
4453:3C 1C 00 00 00 00 F0 FC DF 4683:1C 38 F8 E0 00 E0 E0 C0 5F 48B3:AD F1 CF C9 0B B0 01 60 AB
445B:1C 0E 0E 0E 0E FE FE 85 468B:C0 80 80 00 9C DC FC D1 48BB:A9 00 8D F1 CF 60 EE F1 C1
4463:02 00 00 10 F0 00 FE FE 72 4693:FC 7C 3C 1E 00 80 C0 E0 8B 48C3:CF A9 00 8D F0 CF 60 AD B4
446B:02 00 00 20 E0 00 C0 F8 78 469B:70 38 1C 3E 00 80 80 80 59 48CB:A1 47 C9 01 F0 01 60 EE 83
4473:3C 1C 00 00 00 00 7E 3C 5A 46A3:80 80 80 C0 00 80 80 00 AE 48D3:F3 CF AD F3 CF C9 03 B0 A3
447B:3C 3C 3C 3C FC 00 F8 C0 17 46AB:00 02 FE FE 00 02 97 4C 0C 48DB:01 60 A9 00 8D F3 CF AD C3
4483:C0 C0 C0 C0 C0 00 FE 78 3D 46B3:A0 49 A9 01 8D 19 D0 A9 44 48E3:F2 CF C9 01 F0 16 CE CE 77
448B:78 78 78 78 78 00 3E 1C 61 46BB:1D 8D 18 D0 AD CE 47 8D 0F 48EB:47 AD CE 47 C9 BF F0 01 0A
4493:38 70 E0 C0 80 00 00 00 80 46C3:16 D0 A9 00 20 46 47 8A 16 48F3:60 A9 C0 8D CE 47 A9 01 F7
449B:00 00 00 00 00 00 1E 3C 9C 46CB:A2 3C B9 58 47 EC 12 D0 58 48FB:8D F2 CF 60 EE CE 47 AD FE
44A3:7C FC DC 9C 1C 00 1E 1C 48 46D3:D0 FB 20 46 47 E8 C8 C0 60 4903:CE 47 C9 C8 F0 01 60 A9 8A
44AB:1C 1C 1C 9C DC 00 C0 F8 F7 46DB:3C D0 EF AD A2 47 20 46 46 490B:C7 8D CE 47 A9 00 8D F2 8E
44B3:3C 1C 0E 0E 0E 00 E0 F8 2F 46E3:47 20 4D 47 A0 00 A2 70 F4 4913:CF 60 A0 00 B9 70 43 99 69
44BB:3C 1C 1C 3C F8 00 C0 F8 F2 46EB:B9 88 47 EC 12 D0 D0 FB A0 491B:08 30 B9 40 44 99 08 32 C3
44C3:3C 1C 0E 0E 0E 00 E0 F8 3F 46F3:20 46 47 E8 C8 C0 09 D0 C5 4923:B9 10 45 99 08 34 B9 E0 3E
44CB:3C 1C 1C 3C F8 00 E0 F8 43 46FB:EF AD A3 47 20 46 47 20 9D 492B:45 99 08 36 C8 C0 D0 D0 E6
44D3:3C 3E 00 00 C0 00 FE FE 0D 4703:4D 47 A0 00 A2 CA B9 88 5A 4933:E3 A0 00 D0 00 99 00 31 12
44DB:82 80 80 80 80 00 3E 1C 7A 470B:47 EC 12 D0 D0 FB 20 46 C4 493B:C8 C0 08 D0 F8 60 AD 11 26
44E3:1C 1C 1C 1C 1C 00 3E 1C 40 4713:47 E8 C8 C0 09 D0 EF AD BD 4943:D0 29 7F 8D 11 D0 A9 7F EF
44EB:1C 38 38 70 70 00 3E 1C BA 471B:CD 47 8D 16 D0 AD A4 47 43 494B:8D 0D DC A0 46 A2 B5 8E 44
44F3:1C 1C 1C 1C 00 3E 1C 50 4723:20 46 47 AE 58 47 A0 01 49 4953:14 03 8C 15 03 A9 81 8D E2
44FB:38 70 E0 C0 80 00 3E 1C 81 472B:B9 58 47 99 57 47 C8 C0 59 495B:1A D0 60 78 A9 F0 8D 1A 09
4503:38 70 E0 C0 80 00 FE FE EE 4733:30 D0 F5 8E 87 47 AD 0D 77 4963:D0 A9 31 8D 14 03 A9 EA B2
450B:3C 78 F0 E0 C0 F0 FF FF C7 473B:DC 29 01 F0 03 4C 31 EA 48 496B:8D 15 03 A9 81 8D 0D DC 3E

```



```

4973:20 A3 FD 20 5B FF 58 A9 F5
497B:00 20 46 47 4C 4D 47 A9 23
4983:02 8D FF CF A2 00 CA D0 F2
498B:FD CE FF CF D0 F8 60 A0 99
4993:00 B9 29 4A 99 A7 02 C8 96
499B:C0 50 D0 F5 60 20 15 49 13
49A3:A9 01 8D 86 02 A9 93 20 63
49AB:D2 FF A9 08 20 D2 FF A9 53
49B3:00 8D 15 D0 A0 00 B9 00 D1
49BB:40 99 40 05 B9 EF 40 99 D4
49C3:2F 06 C8 C0 EF D0 EF 20 57
49CB:41 49 20 0E 48 AD 9F 47 B5
49D3:20 78 48 A9 00 20 83 48 08
49DB:AD A0 47 20 6D 48 AD 9E DE
49E3:47 20 91 48 A0 41 A2 E0 09
49EB:86 FE 84 FF A9 00 8D F0 6B
49F3:CF 8D F1 CF 8D F2 CF 8D 72
49FB:F3 CF 20 82 49 20 CF 47 5A
4A03:20 9F 48 20 CA 48 A5 C5 23
4A0B:C9 3C D0 EE 20 5E 49 A9 53
4A13:93 20 D2 FF A9 00 85 C6 F2
4A1B:AD 29 4A C9 02 D0 01 60 6C
4A23:20 92 49 4C B8 02 02 2A 56
4A2B:2A 2A 2A 2A 2A 2A 2A BF
4A33:2A 2A 2A 2A 2A 2A 2A A9 47
4A3B:00 8D 86 02 A9 10 A2 A8 9F
4A43:A0 02 20 BD FF A9 00 A2 D1
4A4B:08 A0 01 20 BA FF 20 D5 1A
4A53:FF AD 29 4A C9 01 F0 03 54
4A5B:4C 00 C0 A9 52 8D 77 02 82
4A63:A9 55 8D 78 02 A9 4E 8D 3C
4A6B:79 02 A9 0D 8D 7A 02 A9 47
4A73:04 85 C6 60 00 00 00 00 4A

```

DEMO MAKER

```

BA 10 REM COPYRIGHT 1992 COMPU
TE PUBLICATIONS INTL LTD
- ALL RIGHTS RESERVED
SA 20 IF A=0 THEN A=1:LOAD"DEM
O.CODE",8,1
HD 30 DIM C$(15),V(9),R(4)
SS 40 FOR X=0 TO 15:READ A$:C$(
X)=A$:NEXT
CG 50 FOR X=0 TO 9:READ A:V(X)
=A:NEXT
PD 60 FOR X=0 TO 44:READ A:POK
E X+32768,A:NEXT
AA 70 R(0)=1:R(1)=2:R(2)=5:R(3
)=6:R(4)=7
RF 80 T$="YES":S=1185:K$="A":X
=0
KS 90 B$="{WHT}-----
-----"
RD 100 POKE53272,23:POKE53281,
0:POKE53280,11
PH 110 PRINT"{CLR}{DOWN}{7}
{10 SPACES}** DEMO MAKE
R **"
QE 120 PRINT"{2 DOWN}
{3 SPACES}{WHT}TITLE LE
TTERS{2 SPACES}COLOR:
{6}"C$(V(0))
AE 130 PRINT"{3 SPACES}{WHT}SC
ROLLING TEXT COLOR: {6}
"C$(V(1))
CE 140 PRINT"{3 SPACES}{WHT}TE
XT ON RASTER COLOR: {6}

```

```

"CS(V(2))
CE 150 PRINT"{3 SPACES}{YEL}MI
D BACKGROUND COLOR: {7}
"CS(V(3))
HD 160 PRINT"{3 SPACES}{YEL}LO
W BACKGROUND COLOR: {7}
"CS(V(4))
DX 170 PRINT"{3 SPACES}{GRN}MA
IN RASTER{2 SPACES}PART
1: {3}"C$(V(5))" SHADE
S"
FJ 180 PRINT"{3 SPACES}{GRN}MA
IN RASTER{2 SPACES}PART
2: {3}"C$(V(6))" SHADE
S"
DG 190 PRINT"{3 SPACES}{GRN}SM
ALL RASTER COLORS:
{PUR}"C$(V(7))" SHADES"
SG 200 PRINT"{3 SPACES}{GRN}MI
DDLE FADER COLORS:
{PUR}"C$(V(8))" SHADES"
QR 210 PRINT"{3 SPACES}{CYN}FA
DER SPEED (0 TO 10):
{PUR}"V(9)
PF 220 PRINT"{3 SPACES}{YEL}AN
IMATE TITLE: {6}"T$
AX 230 PRINT"{3 SPACES}{WHT}EN
TER TITLE TEXT"
XE 240 PRINT"{3 SPACES}ENTER S
CROLLING TEXT"
PD 250 PRINT"{3 SPACES}ENTER M
IDDLE TEXT"
SP 260 PRINT"{3 SPACES}{CYN}SA
VE THE DEMO"
RJ 270 PRINT"{2 DOWN}{6}
{4 SPACES}PRESS SPACE T
O PREVIEW THE DEMO"
BJ 280 SC=S+X*40:POKE SC,62:PO
KESC+54272,1
FD 290 GET C$:IFC$="{DOWN}" AN
D X<14 THEN POKESC,32:X
=X+1
SE 300 IFC$="{UP}"AND X>0 THEN
POKESC,32:X=X-1
GC 310 IF C$ <> "+" THEN 350
FJ 320 IF X<=4 THEN IF V(X)<15
THEN V(X)=V(X)+1:GOTO1
00
DS 330 IF X=9 THEN IF V(X)<10
{SPACE}THEN V(X)=V(X)+1
:GOTO100
DJ 340 IF X<=8 THEN IF V(X)<7
{SPACE}THEN 480
EP 350 IF C$ <> "-" THEN 390
AE 360 IF X<=4 THEN IF V(X)>0
{SPACE}THEN V(X)=V(X)-1
:GOTO100
RA 370 IF X=9 THEN IF V(X)>0 T
HEN V(X)=V(X)-1:GOTO100
XG 380 IF X<=8 THEN IF V(X)>1
{SPACE}THEN 500
JA 390 IF X=10 AND C$=CHR$(13)
THEN 460
HP 400 IF X=11 AND C$=CHR$(13)
THEN 770
FB 410 IF X=12 AND C$=CHR$(13)
THEN 640
GF 420 IF X=13 AND C$=CHR$(13)

```

```

THEN 700
BS 430 IF X=14 AND C$=CHR$(13)
THEN 1100
FP 440 IF C$=" " THEN POKE 189
85,2:GOSUB 940:SYS 1809
8:GOTO 100
XC 450 GOTO 280
RG 460 IFT$="YES" THEN T$="NO"
:GOTO100
FF 470 IFT$="NO" THEN T$="YES"
:GOTO100
SH 480 V(X)=V(X)+1:IF V(X)=3TH
ENV(X)=5
XA 490 GOTO100
KE 500 V(X)=V(X)-1:IF V(X)=4TH
ENV(X)=2
MA 510 GOTO100
PJ 520 REM * TEXT INPUT *
EQ 530 M=0
XS 540 PRINT"{RVS}{WHT} {OFF}"
;
KF 550 GET C$:IFC$=""THEN550
EJ 560 PRINT"{LEFT} {LEFT}";:I
FC$=CHR$(13)THEN RETURN
SB 570 IFC$<>CHR$(20)THEN610
XM 580 IF M=0THEN540
QR 590 M=M-1:PRINT"{LEFT}
{LEFT}";:IFM<1THEN530
RM 600 GOTO540
DE 610 IF C$="{F1}"THEN RETURN
MR 620 IFM>=LE OR (C$<" "ORC$>
"Z") THEN540
GJ 630 M=M+1:PRINTC$;:GOTO540
HH 640 REM * SCROLLING TEXT *
JF 650 PRINT CHR$(147)B$
{10 DOWN}B$"{YEL}
{DOWN}ENTER THE SCROLL
{SPACE}TEXT NOW."
HX 660 PRINT"{DOWN}PRESS <RETU
RN> TO STORE,{2 SPACES}
<F1> TO ABORT{HOME}
{WHT}{DOWN}";
SP 670 FOR Y=16864 TO 17263:PO
KE Y-15800,PEEK(Y):NEXT
:LE=398:GOSUB 520
PX 680 IF C$<>"{F1}"THEN FOR Y
=1064 TO 1463:POKE Y+15
800,PEEK(Y):POKE Y,42:N
EXT
FQ 690 GOTO100
AX 700 REM * MIDDLE TEXT *
MB 710 PRINT CHR$(147)B$
{12 DOWN}B$"{YEL}
{DOWN}ENTER THE MIDDLE
{SPACE}TEXT NOW.
{2 SPACES}FIRST AND
PR 720 PRINT"LAST LINES ARE TE
XT ON RASTERS."
AD 730 PRINT"{DOWN}PRESS <RETU
RN> TO STORE,{2 SPACES}
<F1> TO ABORT{HOME}
{WHT}{DOWN}";
PF 740 FOR Y=16384 TO 16863:PO
KE Y-15320,PEEK(Y):NEXT
:LE=479:GOSUB 520
FE 750 IF C$<>"{F1}"THEN FOR Y
=1064 TO 1543:POKE Y+15
320,PEEK(Y):POKE Y,42:N

```



```

EXT Y
PB 760 GOTO100
RX 770 REM * TITLE *
CR 780 PRINT CHR$(147)BS"
      {2 DOWN}"BS"{YEL}{DOWN}
      ENTER THE TITLE LETTERS
      NOW."
XD 790 POKE1084,45:POKE1124,45
KX 800 PRINT"{DOWN}18 LETTERS
      {SPACE}PER LINE (SPACES
      INCLUDED)"
QD 810 PRINT"NO PUNCTUATION IS
      ALLOWED IN TITLE."
HH 820 PRINT"{DOWN}PRESS <RETU
      RN> AT END OF EACH LINE
      "
XS 830 FOR Y=18341 TO 18360:P=
      Y-17277:V=PEEK(Y):IF V<
      >32 THEN V=V-64
GK 840 POKEP,V:NEXT
PQ 850 FOR Y=18361 TO 18380:P=
      Y-17257:V=PEEK(Y):IF V<
      >32 THEN V=V-64
SS 860 POKEP,V:NEXT
GE 870 PRINT"{HOME}{WHT}{DOWN}
      ";LE=19:GOSUB520
JJ 880 PRINT:LE=19:GOSUB520
PB 890 FOR Y=1064 TO 1083:P=Y+
      17277:V=PEEK(Y):IF V<>3
      2 THEN V=V+64
QD 900 POKEP,V:POKE Y,42:NEXT
KC 910 FOR Y=1104 TO 1122:P=Y+
      17257:V=PEEK(Y):IF V<>3
      2 THEN V=V+64
ER 920 POKEP,V:POKE Y,42:NEXT
BM 930 GOTO100
PQ 940 REM * CUSTOMIZER *
HP 950 PRINT"{CLR}"
PP 960 AD=18333:POKE AD,V(9):P
      OKE AD+1,V(2):POKE AD+2
      ,V(0):POKE AD+3,V(1)
BG 970 POKE AD+4,0:IF TS="YES"
      THEN POKE AD+4,1
PQ 980 POKE AD+6,V(3):POKE AD+
      7,V(4)
BP 990 REM * RASTERCOL *
FK 1000 Y=32768:FOR T=0 TO 3:I
      F R(T)<>V(7) THEN Y=Y+
      9:NEXT
XC 1010 A=18312:GOSUB1090
SR 1020 Y=32768:FOR T=0 TO 3:I
      F R(T)<>V(8) THEN Y=Y+
      9:NEXT
EB 1030 A=18321:GOSUB1090
HM 1040 Y=32768:FOR T=0 TO 3:I
      F R(T)<>V(5) THEN Y=Y+
      9:NEXT
MR 1050 A=18264:FOR T=Y TO Y+7
      :FOR U=0 TO 2:POKE A,P
      EEK(T):A=A+1:NEXT U:NE
      XT T
AS 1060 Y=32768:FOR T=0 TO 3:I
      F R(T)<>V(6) THEN Y=Y+
      9:NEXT
AP 1070 A=18288:FOR T=Y TO Y+7
      :FOR U=0 TO 2:POKE A,P
      EEK(T):A=A+1:NEXT U:NE
      XT T

```

```

XF 1080 RETURN
AB 1090 FOR T=Y TO Y+9:POKE A,
      PEEK(T):A=A+1:NEXT:RET
      URN
QE 1100 REM{2 SPACES}* SAVE *
KJ 1110 GOSUB 940:INPUT"{CLR}
      {WHT}{DOWN}SHOULD DEMO
      BOOT ANOTHER FILE? Y
      {3 LEFT}";AS
AB 1120 IF AS="N" THEN POKE 18
      985,2:GOTO 1200
EX 1130 PRINT"{CLR}{2 DOWN}"BS
      "{YEL}ENTER THE NAME O
      F FILE TO BOOT"
CF 1140 PRINT"(TYPE '*' AT END
      OF NAME){HOME}":POKE1
      080,45:LE=15:GOSUB520
XM 1150 FOR Y=1064 TO 1079:A=Y
      +17922:V=PEEK(Y):IF V
      {SPACE}> 0 AND V < 27
      {SPACE}THEN V=V+64
AQ 1160 POKE A,V:NEXT:PRINT
XF 1170 INPUT"{4 DOWN}{CYN}ENT
      ER SYS (RETURN=BASIC)"
      ;SY
DD 1180 IF SY=0 THEN POKE18985
      ,1:GOTO 1200
SC 1190 POKE 18985,0:H=INT (SY/
      256):L=SY-256*H:POKE 1
      9036,L:POKE 19037,H
MF 1200 INPUT"{CLR}{2 DOWN}
      {WHT}FILENAME OF DEMO"
      ;FS
PK 1210 PRINT"{2 DOWN}{GRN}SAV
      ING DEMO...";
BG 1220 OPEN2,8,2,"0:"+"FS+",P,
      W"
XH 1230 PRINT#2, CHR$(1);CHR$(
      8);
HB 1240 FOR J=16252 TO 19064:P
      RINT#2,CHR$(PEEK(J));
CS 1250 IF ST > 0 THEN PRINT"
      {SPACE}ERROR DURING SA
      VE":PRINT#2:CLOSE2:END
      NEXT:CLOSE2:PRINT" NO
      {SPACE}ERRORS":END
PG 1260
AM 1270 DATA BLACK,WHITE,RED,C
      YAN,PURPLE,GREEN,BLUE,
      YELLOW
HA 1280 DATA ORANGE,BROWN,LT R
      ED,DK GRAY,MED GRAY,LT
      GREEN,LT BLUE,LT GRAY
JP 1290 DATA 0,1,11,0,5,1,2,6,
      1,5
BK 1300 DATA 0,11,12,15,1,1,15
      ,12,11
SF 1310 DATA 0,9,2,8,10,10,8,2
      ,9
KF 1320 DATA 0,5,3,13,1,1,13,3
      ,5
XD 1330 DATA 0,6,14,3,1,1,3,14
      ,6
FF 1340 DATA 10,7,13,3,14,3,13
      ,7,10

```

F/X PLOT 128

By Kevin Davis

At one time or another, any student of mathematics has had to deal with coordinate plane graphing. Graphing calculators have been recognized by educators as helpful tools for understanding graphs, but these calculators can cost up to \$300.

While programs such as Arbplot exist for Apple and IBM computers, few are available for 8-bit Commodores. F/X Plot 128 tries to change this. The program itself was designed using the popular TI-81 as a benchmark. It's a menu-driven, user-friendly, colorful program designed to make good use of the 128's high-resolution graphics in 40-column mode.

Entering the Program

F/X Plot 128 consists of a main program and a sprite. The main program is written in BASIC 7.0 and should be entered with The Automatic Proofreader to avoid typing errors. See "Typing Aids" elsewhere in this section.

The sprite data used to create an on-screen pointer or tracer is written in machine language. Enter it with MLX, our machine language entry program. See "Typing Aids" again. When MLX prompts, respond with the following values.

Starting address: 0E00

Ending address: 0E3F

Be sure to save the program as F/X PLOT.SPR on the same disk as the main program. The main program loads this data when it runs.

When you first run F/X Plot, the top of the screen will be jumbled. Clear the screen from the Options menu, press Run/Stop-Restore, and then run the program again. This process is required only when you first run the program.

From then on, you'll see a screen divided into quarters with a menu at the bottom. The program is completely menu driven. First, let's look at the Options menu. Press 4.

With this menu, you'll be presented with five numbered options. Press the number in front of the one you want.

Simultaneous/Sequence F/X Plot can hold formulas for up to four graphs.

Danny English is a frequent contributor who lives in Moreno Valley, California.

Simultaneous will graph all selected formulas at the same time. Sequence graphs them one at a time. Always select Sequence when you have only one formula to graph. (Sequence is also a little faster in this case.)

Enable/Disable Formulas You can toggle here to enable or disable certain formulas. Only the enabled ones, denoted by their numbers appearing in reverse mode, will be graphed.

Clear Graphics Screen This will clear the graphics screen and return you to the Main menu.

Set Tracer Speed This option allows you to set the speed at which the tracer moves. A smaller number will move more slowly but will be more accurate. Three is the default.

Connected/Dotted This works in a fashion similar to the Enable/Disable toggle. The formulas set to Connected will draw a line between each plotted point. Experimentation with the program will give you an idea of when it's a good idea to set formulas to Connected. Connected formulas are in reverse.

Use the Esc key to move from the Options screen to the Main menu. Here are the Main menu options.

Set Range Set the X-Minimum (the least x value graphed), X-Maximum (the greatest x value graphed), Y-Minimum, and Y-Maximum. Defaults are -10 to 10 on both axes. The resolution (scaling) of the graph is determined by the distance between the maximum and minimum values. In other words, a range of -1 to 1 will appear much closer in (like a zoom lens) and have a higher resolution than -100 to 100. For trigonometric graphs, use plus or minus 3.14 as pi and plus or minus 6.28 as two pi.

Tracer The Tracer requires a joystick in port 2. When you choose this, you'll see the range values and a gray device somewhat like cross hairs in the upper left corner of the screen. Its speed is controlled from the Options menu. You're shown the x-coordinate and y-coordinate of the cross hairs as they

move around the screen. This option can be helpful in evaluating certain points along the graph.

In order to zoom into a section of the graph, move to the upper left corner of the rectangular section you want to zoom into and press and release the fire button. There will be a bell tone and a brief pause. Then move the cross hairs to the lower right corner and do likewise. This process sets the next range of values. Use the Esc key to exit, select Clear from the Options menu, and choose Graph from the Main menu to proceed.

Graph This is the most important, and easiest, function of F/X Plot. It graphs the enabled functions on the screen. Each function is graphed in its corresponding color as shown in the Enable/Disable menu. As each point is graphed, its x and y values are printed at the bottom of the screen. The coordinates of the last five points graphed are visible at the bottom of the screen. To pause the graphing, use the No Scroll key.

If the program attempts to graph a value that BASIC cannot handle (such as the point on an asymptote in a trigonometric function), a bell tone will sound, and that point won't be graphed. An error-trapping routine prevents the program from stopping. It'll just move on to the next value.

Plotting Functions

F/X Plot comes with four formulas already entered, but, naturally, you'll want to enter your own. These are stored as DEF FN statements in lines 60-90. The BASIC DEF FN command, however, cannot be implemented from program mode. In order to set or change the formulas, you must exit the program by pressing the Run/Stop-Restore keys simultaneously. Type LIST 60-90 and press Return.

Then use the cursor keys to edit each line with the formula of your choice for FN(A), FN(B), FN(C), and FN(D). Remember to press Return after entering your formulas. Then run the program again.

All formulas must be entered using BASIC's normal mathematical commands. These include SIN, COS, LOG, TAN, ABS, and operators +, -, *, and /.

Anyone who has ever done any programming should be familiar with these operators and their use. New users may want to check the appendix dealing with mathematical commands in the *128 System Guide* for the proper format.

Helpful Hints

While the program can graph only four functions at once, the graphics screen isn't erased when you stop the program. If you want to graph more than four formulas, graph the first four, exit the program, change the formulas, run the program again, and graph the second group. The graphs will overlap as long as you make sure that the ranges are the same.

Don't wait until the night before your big geometry examination to learn how to use F/X Plot 128! It's a powerful and versatile program that can be utilized fully only by thorough use and repeated experimentation.

While the theory is untested, I'd imagine that any program that captures the hi-res split screen and sends it to a printer could be used to make hardcopies of any graphs you create.

F/X PLOT 128

```
RP 5 REM COPYRIGHT 1992 COMPUT
E PUBLICATIONS INTL LTD
{2 SPACES}ALL RIGHTS RESE
RVED
HX 10 CLR:XM=-10:XX=10:YM=-10:
YX=10:SP=3:EA=1:EB=1:EC=
1:ED=1:TRAP 1890
SA 20 DIMJA(8):DIMJB(8):FORKQ=
1TO8:READJA(KQ):NEXT
RK 30 BLOAD("F/X PLOT.SPR")
PJ 40 FOR KQ=1TO8:READJB(KQ):N
EXT
MA 50 POKE53280,0:POKE53281,0
AQ 60 DEF FN A(X)=X↑2
PF 70 DEF FN B(X)=X+4
EJ 80 DEF FN C(X)=((SIN(X)))
QK 90 DEF FN D(X)=(-1*(SQR(X)))
MA 100 GRAPHIC 4,0
MM 110 REM DRAW ZEROS
KS 120 IF XM<0 AND XX>0 THEN X
Z=1
RP 130 IF YM<0 AND YX>0 THEN Y
Z=1
RR 150 GOSUB 840
FR 160 SLEEP1:GOTO890
MB 200 GOSUB 770
CG 210 IF XZ<>1 THEN 270
SM 220 REM DRAW X-ZERO
DG 230 AM=ABS(XM):XL=AM/XR
```


PROGRAMS

```

DA 240 XL=160*XL
BE 250 COLOR 1,16
GG 260 DRAW 1,XL,0 TO XL,159
FH 270 IF YZ<>1 THEN 320
KR 280 YQ=ABS(YM):YU=YQ/YR
JK 290 YL=160*YU:YL=160-YL
HB 300 IF YL>160 OR YL<0 THEN
{SPACE}320
DB 310 COLOR2,16:DRAW1,0,YLTO1
59,YL
EG 320 REM PLOT
EE 330 XS=XR/160:YS=YR/160
MH 340 UX=0:UY=0
FD 350 FOR TT=0TO159
JJ 360 IF EA=0 THEN 460
JX 370 XA=FNA((XM+(TT*XS)))
BA 380 PRINT"{CYN}X="((XM+(TT*
XS)))TAB(20)"Y="XA
RK 390 IF XA>YX OR XA<YM THEN
{SPACE}450
XX 400 QP=YX-XA:FP=QP/YR:CP=16
0*FP
EK 410 COLOR 1,4
BD 420 DRAW 1,TT,CP
RH 430 IF TT=0 OR CA<>1 THEN45
0
AS 440 DRAW1,TT,CP TO IA,PA:IA
=TT:PA=CP
KF 450 IF SS=1 THEN NEXT TT
AB 460 IF EB=0 THEN 570
PH 470 IF SS=1 THEN UX=0:UY=0:
FOR TT=0 TO 159
MH 480 XB=FNB((XM+(TT*XS)))
SS 490 PRINT"{GRN}X="((XM+(TT*
XS)))TAB(20)"Y="XB
MD 500 IF XB>YX OR XB<YM THEN
{SPACE}560
JH 510 QP=YX-XB:FP=QP/YR:CP=16
0*FP
SB 520 COLOR1,6
AJ 530 DRAW 1,TT,CP
EM 540 IF TT=0 OR CB<>1 THEN56
0
RR 550 DRAW1,TT,CP TO IB,PB:IB
=TT:PB=CP
SS 560 IF SS=1 THEN NEXT TT
KJ 570 IF EC=0 THEN 660
KS 580 IF SS=1 THEN UX=0:UY=0:
FORTT=0TO159
PX 590 XC=FNC((XM+(TT*XS)))
DB 600 PRINT"{YEL}X="((XM+(TT*
XS)))TAB(20)"Y="XC
HM 610 IF XC>YX OR XC<YM THEN
{SPACE}650
EF 620 QP=YX-XC:FP=QP/YR:CP=16
0*FP:COLOR1,8:DRAW1,TT,
CP
AA 630 IF TT=0 OR CC<>1 THEN65
0
G 640 DRAW1,TT,CP TO IC,PC:IC
=TT:PC=CP
GB 650 IF SS=1 THEN NEXT TT
KR 660 IF ED=0 THEN 740
GD 670 IF SS=1 THEN UX=0:UY=0:
FORTT=0TO159
PF 680 XD=FND((XM+(TT*XS)))
SR 690 PRINT"{3}X="((XM+(TT*XS
)))TAB(20)"Y="XD
FD 700 IF XD>YX OR XD<YM THEN
{SPACE}740
DF 710 QP=YX-XD:FP=QP/YR:CP=16
0*FP:COLOR1,11:DRAW1,TT
,CP
XB 720 IF TT=0 OR CQ<>1 THEN74
0
JC 730 DRAW1,TT,CP TO ID,PD:ID
=TT:PD=CP
AE 740 IF EA=1 OR EB=1 OR EC=1
OR ED=1 OR SS=0 THEN N
EXT TT
AP 750 IF EA=0 AND EB=0 AND EC
=0 AND ED=0 THEN 760
EJ 760 GOTO890
BC 770 IF XX<=0 AND XM<=0 THEN
XR=ABS(XM)-ABS(XX)
BF 780 IF XX>0 AND XM<0 THEN X
R=ABS(XM)+ABS(XX)
MG 790 IF XM>0 THEN XR=XX-XM
AM 800 IF YX<=0 AND YM<=0 THEN
YR=ABS(YM)-ABS(YX)
HR 810 IF YX>0 AND YM<0 THEN Y
R=ABS(YM)+ABS(YX)
AD 820 IF YM>0 THEN YR=YX-YM
GP 830 RETURN
FP 840 REM PIC
FE 850 PRINT"{CLR}{20 DOWN}";
XP 860 PRINT"{YEL}{40 @}"
DJ 870 PRINT"{YEL}{16 @}{3}F
{CYN}/{7}X {6}PLOT{YEL}
{16 @}"
CX 880 RETURN
BD 890 GOSUB840:PRINT" {6}
{RVS}{H}1{N}{OFF} SET R
ANGE VALS{2 SPACES}
{RVS}{H}2{N}{OFF} USE T
RACER"
AH 900 PRINT" {RVS}{H}3{N}
{OFF} GRAPH FUNCTION
{2 SPACES}{RVS}{H}4{N}
{OFF} OPTIONS"
DB 910 GETKEY GK$:GK=VAL(GK$):
IF GK>4 OR GK<1 THEN 91
0
RG 920 ON GK GOTO 930,1450,170
,990
PG 930 REM SET VALUES
MX 940 INPUT"{CLR}{20 DOWN}X-M
INIMUM [XMIN]";XM
KB 950 INPUT"X-MAXIMUM [XMAX]"
;XX
MD 960 INPUT"Y-MINIMUM [YMIN]"
;YM
QG 970 INPUT"Y-MAXIMUM [YMAX]"
;YX
GM 980 PRINT"{CLR}":GOTO890
PD 990 REM OPTIONS MENU
MJ 1000 PRINT"{CLR}{19 DOWN}
{8 SPACES}{7}
{20 SPACES}"
QP 1010 PRINT"{RVS}{H}1{N}
{OFF} SIMULTANEOUS/SEQ
UENCE{3 SPACES}{3}OPTI
ONS{7}"
XM 1020 PRINT"{RVS}{H}2{N}
{OFF} ENABLE/DISABLE F
ORMULAS{3 SPACES}{3}ME
NU{7}{2 SPACES}"
HX 1030 PRINT"{RVS}{H}3{N}
{OFF} CLEAR GRAPHIC SC
REEN"
EF 1040 PRINT"{RVS}{H}4{N}
{OFF} SET TRACER SPEED
"
GM 1050 PRINT"{RVS}{H}5{N}
{OFF} CONNECT/DOTTED";
XX 1060 GETKEY GK$:IFGK$=CHR$(
27) THEN 890
PK 1070 GK=VAL(GK$):IF GK>5ORG
K<1THEN1060
BS 1080 ON GK GOTO 1090,1160,1
370,1400,1690
RB 1090 PRINT"{CLR}{20 DOWN}
{GRN}1] SEQUENCE (GRAP
HED IN ORDER)"
PR 1100 PRINT"2] SIMULTANEOUS
{SPACE}(GRAPHED AT SAM
E TIME)"
SS 1110 GETKEY GK$:GK=VAL(GK$)
:IFGK=1THENSS=1
GQ 1120 IF GK=2 THEN SS=0
CF 1130 IF GK$=CHR$(27) THEN 8
90
FP 1140 IF SS<0ORSS>1THEN1110
HM 1150 GOTO890
MR 1160 PRINT"{CLR}{20 DOWN}
{1}PRESS FORMULA TO EN
ABLE (HI-LIGHTED)"
JS 1170 PRINT"OR DISABLE.
{RVS}RETURN{OFF} EXITS
."
KH 1180 IF EA=1THENPRINT"{7}
{RVS}{G}1{N}{OFF}";:GO
TO1200
JA 1190 IF EA<>1 THEN PRINT"
{7} 1 ";
EX 1200 IF EB=1THENPRINT"{GRN}
{RVS}{H}2{N}{OFF}";:GO
TO1220
PP 1210 IF EB<>1 THEN PRINT"
{GRN} 2 ";
RS 1220 IF EC=1THENPRINT"{YEL}
{RVS}{H}3{N}{OFF}";:GO
TO1240
KH 1230 IF EC<>1 THEN PRINT"
{YEL} 3 ";
PQ 1240 IF ED=1THENPRINT"{3}
{RVS}{H}4{N}{OFF}";:GO
TO1260
CQ 1250 IF ED<>1 THEN PRINT"
{3} 4 ";
QX 1260 GETKEY CD$:IFCD$=CHR$(
13)THEN890
JR 1270 CD=VAL(CD$):IFCD>4ORCD
<1THEN1260
BB 1280 IF CD=1 AND EA=0 THEN
{SPACE}EA=1:GOTO1300
EP 1290 IF CD=1 AND EA=1 THEN
{SPACE}EA=0
BG 1300 IF CD=2 AND EB=0 THEN
{SPACE}EB=1:GOTO1320
KE 1310 IF CD=2 AND EB=1 THEN
{SPACE}EB=0
SB 1320 IF CD=3 AND EC=0 THEN
{SPACE}EC=1:GOTO1340

```



```

DA 1330 IF CD=3 AND EC=1 THEN
{SPACE}EC=0
PJ 1340 IF CD=4 AND ED=0 THEN
{SPACE}ED=1:GOTO1360
RX 1350 IF CD=4 AND ED=1 THEN
{SPACE}ED=0
JS 1360 PRINT "{UP}":GOTO 1180
FP 1370 PRINT "{CLR}{22 DOWN}"
{WHT}CLEAR SCREEN: ARE
YOU SURE?"
SC 1380 GETKEY YN$:IF YN$="Y" T
HENGGRAPHIC4,1
GM 1390 GOTO 890
CG 1400 PRINT "{CLR}{21 DOWN}"
{PUR}"
CM 1410 PRINT "CURRENT SPEED:
{WHT}"SP:PRINT "{PUR}EN
TER NEW VALUE [1-5]"
CD 1420 GETKEY SP$:IF SP$=CHR$(
27) THEN 890
KH 1430 SP=VAL(SP$):IF SP<10RS
P>5 THEN 1420
RQ 1440 GOTO 890
PR 1450 REM TARGET-SCAN
EC 1460 ZX=XM:ZY=YX
DM 1470 PX=17:PY=-209:PN=0
SX 1480 PRINT "{CLR}{33 DOWN}"
FK 1490 PRINT "{YEL}X-MIN:"XM"
TAB(20)"X-MAX:"XX:PRIN
T"Y-MIN:"YM"TAB(20)"Y
-MAX:"YX
HE 1500 SPRITE1,1,2,0
RK 1510 MOVSPR1,PX,PY
KD 1520 GOSUB 770
BC 1530 XS=XR/160:YS=YR/160
PQ 1540 JS=0:JS=JOY(2)
DH 1550 IF JS>128 THEN 1540
XP 1560 IF JS=128 THEN 1660
PM 1570 PX=PX+(JA(JS)*SP):PY=P
Y+(JB(JS)*SP)
KD 1580 MOVSPR1,PX,PY
GD 1590 ZX=XM+(.5*((PX-17)*XS)
):ZY=YX-((PY+209)*YS)
CM 1600 PRINT "{6}{HOME}"
{21 DOWN}X="ZX"TAB(20)
"Y="ZY)
KF 1610 GETXS:IFXS=CHR$(27)THE
NGOTO1630
QB 1620 JS=0:GOTO1540
GJ 1630 SPRITE1,0
CP 1640 GOTO 890
MD 1650 DATA 0,1,1,1,0,-1,-1,-
1,-1,-1,0,1,1,0,-1
FQ 1660 PN=PN+1:IF PN=1 THEN A
X=ZX:AY=ZY:PRINT CHR$(
7):SLEEP1:GOTO1540
AA 1670 IF PN=2 THEN BX=ZX:BY=
ZY:PRINT CHR$(7):XM=AX
:XX=BX:YX=AY:YM=BY:GOT
O1540
FQ 1680 GOTO1540
CS 1690 PRINT "{CLR}{20 DOWN}"
{PUR}SELECT DOTTED OR
{SPACE}CONNECTED /
{RVS}RETURN{OFF}"
DD 1700 IF CA=1 THEN PRINT "{7}"
{RVS}{H}1{N}{OFF}";:GO
TO1720

```

```

QD 1710 IF CA<>1 THEN PRINT"
{7} 1 ";
GX 1720 IF CB=1 THEN PRINT "{GRN}"
{RVS}{H}2{N}{OFF}";:GO
TO1740
HP 1730 IF CB>1 THEN PRINT"
{GRN} 2 ";
KS 1740 IF CC=1 THEN PRINT "{YEL}"
{RVS}{H}3{N}{OFF}";:GO
TO1760
BE 1750 IF CC>1 THEN PRINT"
{YEL} 3 ";
HF 1760 IF CQ=1 THEN PRINT "{3}"
{RVS}{H}4{N}{OFF}";:GO
TO1780
XH 1770 IF CQ>1 THEN PRINT"
{3} 4 ";
FG 1780 GETKEY SQ$:IFSQ$=CHR$(
13) THEN 890
HC 1790 SQ=VAL(SQ$):IFSQ>4 OR SQ
<1 THEN 1780
MG 1800 IF SQ=1 AND CA=0 THEN
{SPACE}CA=1:GOTO1820
FD 1810 IF SQ=1 AND CA=1 THEN
{SPACE}CA=0
XB 1820 IF SQ=2 AND CB=0 THEN
{SPACE}CB=1:GOTO1840
DS 1830 IF SQ=2 AND CB=1 THEN
{SPACE}CB=0
AK 1840 IF SQ=3 AND CC=0 THEN
{SPACE}CC=1:GOTO1860
JF 1850 IF SQ=3 AND CC=1 THEN
{SPACE}CC=0
JC 1860 IF SQ=4 AND CQ=0 THEN
{SPACE}CQ=1:GOTO1880
GG 1870 IF SQ=4 AND CQ=1 THEN
{SPACE}CQ=0
AR 1880 PRINT "{UP}":GOTO 1700
ER 1890 REM ERROR TRAP ROUTINE
XA 1900 IF ER=11 OR ER=14 OR E
R=20 OR ER=10 OR ER=15
THEN PRINT "{G}";:RESU
ME NEXT
HH 1910 PRINT "PROGRAMMING ERRO
R:"
AC 1920 PRINT "{RVS}";:PRINTERR
S(ER);:PRINT "{OFF}"
AX 1930 PRINT "{8}IN LINE"EL
CK 1940 END

```

F/X PLOT.SPR

```

0E00:02 00 00 0A 80 00 28 A0 B2
0E08:00 A0 28 00 28 A0 00 0A 1F
0E10:80 00 02 00 00 00 00 00 AC
0E18:00 00 00 00 00 00 00 00 34
0E20:00 00 00 00 00 00 00 00 3C
0E28:00 00 00 00 00 00 00 00 44
0E30:00 00 00 00 00 00 00 00 4C
0E38:00 00 00 00 00 00 00 00 54

```

Kevin Davis is the author of Sport Card Collector (November 1991). F/X Plot 128 helped him survive honors geometry this year at Trinity Preparatory School in Winter Park, Florida.

SPRINT III

By Farid Ahmad

Commodore BASIC isn't exactly known for its speed. It's not the language itself that's at fault, however; it's the way the computer implements the language. BASIC is interpreted. Most faster languages are compiled.

Before a computer can execute any program, the commands must be converted into the machine language of the computer. Every time a statement in an interpreted language is run, the computer must translate it. This slows execution considerably. A compiler, however, translates the entire program before it runs. This boosts execution speed.

Sprint III is a BASIC compiler that supports a subset of BASIC statements available to the 64. Earlier versions of the program were published in January 1986 and January 1988. Sprint II added several features to its original version, but there was still room for improvement.

Entering the Program

Sprint III is written entirely in machine language. To enter it, use MLX, our machine language entry program. See "Typing Aids" elsewhere in this section. When MLX prompts, respond with the following values.

Starting address: 0801

Ending address: 1D40

Be sure to save a copy of Sprint III before exiting MLX.

Using the Program

A typical session with Sprint III might start with the following commands.

LOAD "Sprint III",8,1

NEW

SYS32768

At this point your screen will turn black and BASIC pointers will be adjusted. You'll also see a SYNTAX ERROR message, which you should ignore, and a READY prompt. Now you can load a BASIC program or start to write one from scratch. You may run your program at any time for testing. When you are ready to compile it, use either the SYS32768,E or SYS32768,F command. These and other Sprint III commands

are explained below. When the compilation finishes, you can run your compiled program or save it as usual.

If you want to make some corrections or modifications to your program, use the **SYS32768,L** command to load your BASIC program from the RAM disk. After it's edited, the program can be compiled as before.

Commands

All Sprint III commands consist of a **SYS** statement followed by a comma and a single letter. The following commands are available.

SYS32768,F Saves the BASIC program to the RAM disk and then compiles it. The entire source program is listed during compilation. It also locks the RAM disk.

SYS32768,E Same as above except that Sprint lists only lines containing an error.

SYS32768,L Loads your source program after a compile operation. Unlocks the RAM disk.

SYS32768,U Unlocks the RAM disk.

A Better Way

One of the advantages of using Sprint II was speed. You could save the source code on disk; run Sprint II; and get a compiled program in memory, ready to run. However, if you wanted to make corrections or modifications to your program, you had to load the source code, modify it, save it back to disk, and compile again. Since a program almost never works perfectly without a debugging session, this procedure soon became tedious.

There's a better way. Sprint III turns Sprint into a RAM-based compiler. Now, when you compile your program, the source code is saved to a RAM disk and is compiled from there. When compilation is finished, you have two versions of your program in memory: the compiled program in the normal BASIC working area and the source program in the RAM disk.

Suppose Sprint III has reported some errors during compilation or you simply want to make some modifications. You issue the load command to

the RAM disk, and within seconds, your BASIC source program is back, ready for editing and recompiling.

Speed

Apart from the convenience of not having to save and load repeatedly, Sprint III also reduces compilation time by more than half. For even faster compiling, you're given the option of listing the entire BASIC program during compilation or listing only those lines that contain an error. If you choose the error-only option, time for compiling is reduced to less than one-fourth that of Sprint II.

Other Options

A list pause has been provided. When a BASIC program is being listed to the screen during compilation, hold down any control key to freeze the display. This is useful for writing down any errors reported by Sprint III.

One potential mistake that can occur when using Sprint III is the one of accidentally compiling a program that's already compiled. Suppose that you want to load the source program but give the compile command by mistake. The BASIC program in the RAM disk would be overwritten by the compiled program, and you'd lose your source code. To prevent this, the RAM disk is automatically locked after each compilation. This prevents you from using the compile command. It'll be unlocked automatically when you load your source program, or you can use the unlock command if you wish to start a new program.

Expressions

Sprint III works much the same as BASIC, but as with Sprint II there are some differences you should keep in mind when programming.

Strings are limited to ten characters unless specified otherwise. If you want to include more characters, dimension the string as if it were an array.

In numeric expressions, parentheses aren't allowed. Unlike interpreted BASIC, expressions are evaluated strictly from left to right; there's no operator precedence. To get around this, you must break up the expression into several smaller expressions and then put the results back together. For example,

the following expression:

$$\text{SUM} = A + X*Y + B + I*J$$

becomes

$$\begin{aligned} Q1 &= X*Y; Q2 = I*J; \\ \text{SUM} &= A + Q1 + B + Q2. \end{aligned}$$

You may get a NOT SUPPORTED error message if the statement is legal in Commodore BASIC but illegal in Sprint III. You may get a SYNTAX ERROR message if a keyword is misspelled or extra parentheses are used.

Memory Organization

Memory locations \$A000-\$C7FF and \$E000-\$FFFF are used by the RAM disk. Since this is an 18K area of memory, the BASIC program must fit within this length. You'll be warned if you exceed this limit. A full 30K, however, is available for the compiled program. Also, if you run your BASIC source program for testing, a full 30K of working area is available for strings, variables, and so on. It's only the BASIC program text that shouldn't exceed 18K.

Sprint III is designed to stay resident in memory while you edit your BASIC programs. Therefore, it moves down the top of the BASIC area pointer to protect itself. To set this pointer, you can either use a Sprint III command before running a BASIC program or use **SYS32768** without a letter. The second method will produce a SYNTAX ERROR message, but the pointer will be adjusted. Also, when the pointer is set, the list pause will become available for BASIC editing.

Updates from Sprint II

Programmers who may have disassembled Sprint II may be interested in the following changes.

The address \$8000 now jumps to \$AB00 where the new routines reside. The interface routine sets some pointers and also the colors. It then looks for a comma followed by a letter. If a valid command is found, the appropriate routines are called to move programs to or from the RAM disk.

In Sprint II, addresses \$825E and \$80A2 called a subroutine to initialize the disk file. They now call a new short

routine at \$802B which initializes the new input from the RAM disk routine.

The area \$8C0D-\$8C45 contained calls to a Kernal routine to get a byte from disk. These calls have been changed so that the new input from the RAM disk routine is called at \$8815.

The instruction at \$C800 has been changed from LDA#\$C0 to LDA#\$C8. As a result, Sprint now uses the memory area \$C800 onward for line number storage.

When the E command is used, the JSR\$8AAC instruction at \$82B6 address is bypassed, preventing each line from being listed. Also the instruction at \$8A85 is changed from JSR\$888C to JSR\$8850. Address \$8850 contains a short routine to print the current line.

When Sprint III is used for the first time, it places the list freeze subroutine at \$CFEC-\$CFFF. I use Sprint III with a multifunction cartridge that causes problems if the freeze routine is placed between \$8000 and \$A000.

SPRINT III

```
0801:0B 08 70 17 9E 32 34 30 6E
0809:37 00 00 00 20 20 20 96
0811:20 20 20 20 20 A0 C4 B9 06
0819:3C 08 99 F8 00 B9 FD 08 F6
0821:99 33 03 88 D0 F1 A0 09 4C
0829:B9 0C 08 98 FF 03 88 D0 A1
0831:F7 A9 D1 85 2D A9 24 85 18
0839:2E 4C 00 01 16 60 06 D1 93
0841:24 B8 1B B9 6E 09 99 E8 44
0849:07 C8 D0 F7 EE 02 01 EE 19
0851:05 01 C6 F9 D0 ED A2 03 23
0859:20 34 03 F0 33 C9 07 D0 95
0861:16 A2 01 20 34 03 D0 0A A0
0869:A2 04 20 34 03 18 69 07 65
0871:10 05 A2 0A 20 34 03 85 1D
0879:A8 A5 A7 85 A9 A5 FE 85 FB
0881:F7 A5 FF 85 F8 20 6C 03 73
0889:A5 F8 85 FF A5 F7 85 FE 72
0891:E8 20 34 03 D0 1E A2 08 21
0899:20 34 03 A0 02 84 A8 85 2A
08A1:A6 18 A5 FC 65 A6 85 F7 58
08A9:A5 FD 65 A7 85 F8 20 6C EF
08B1:03 4C 13 01 E8 20 34 03 FB
08B9:D0 1C A0 03 84 A8 E8 20 36
08C1:34 03 F0 08 A2 08 20 34 F4
08C9:03 4C 5C 01 A2 0D 20 34 C7
08D1:03 E6 A7 4C 5C 01 E8 20 AF
08D9:34 03 D0 0A E8 20 34 03 B2
08E1:18 69 04 A8 D0 D6 E8 20 37
08E9:34 03 D0 0A A2 02 20 34 21
08F1:03 18 69 06 D0 ED A2 08 A2
08F9:20 34 03 D0 E6 A9 00 85 F7
0901:A7 A4 FB F0 0C 06 FA 2A 37
0909:26 A7 C6 FB CA D0 F2 A8 D8
```

```
0911:60 48 B1 FE 85 FA A9 08 FE
0919:85 FB 68 A4 FE D0 02 C6 4A
0921:FF C6 FE C0 E7 D0 DE A4 B5
0929:FF C0 07 D0 D8 A9 37 85 BA
0931:01 58 4C 00 80 A4 A8 F0 3C
0939:22 A5 F7 38 E5 A8 B0 03 7E
0941:C6 F8 38 85 F7 A5 FC E5 8A
0949:A8 B0 02 C6 FD 85 FC B1 3A
0951:F7 88 91 FC 98 D0 F8 C4 42
0959:A9 F0 0A B1 F7 C6 FD C6 76
0961:F8 C6 A9 10 EC 60 78 E6 98
0969:01 4C 16 08 60 00 4C 00 ED
0971:9A EA 9D C6 3A A9 3A 69 02
0979:09 E3 27 A9 81 CA 12 A9 8D
0981:FE 85 42 19 43 D6 C0 44 89
0989:34 7C 4C 4F 55 CF C1 18 D8
0991:8D 32 88 A9 A0 8D 33 88 CC
0999:60 A9 38 8D 51 EE 8E 5E 6F
09A1:60 AD 3A 00 E4 E0 99 EE 63
09A9:6A 88 4C 57 CF 86 E3 3A AC
09B1:34 3B 84 F7 A9 08 85 F8 2E
09B9:A9 EE E0 85 83 92 B9 11 BB
09C1:70 3C 0B 2A 1C BB 82 48 3C
09C9:C9 97 50 44 02 C9 D5 90 68
09D1:E9 A9 FF 85 F9 87 D3 A9 DA
09D9:8B FA 85 07 82 31 2E 8D 54
09E1:FF 7D 94 AF 80 18 85 1F 6F
09E9:3B 12 38 A9 C8 85 39 EC 7B
09F1:A1 20 76 81 85 07 30 82 EE
09F9:A5 17 2A E6 17 E6 54 18 6D
0A01:34 B3 40 38 67 16 38 A5 51
0A09:38 E0 C9 0C 38 9C 5E 39 95
0A11:C0 00 03 04 7A 80 01 04 02
0A19:D3 86 2B 05 86 82 25 1F 16
0A21:C9 83 D0 5C 58 DF 44 D0 76
0A29:0C E6 44 5C 2D E3 0B 40 9A
0A31:69 01 E7 06 4F 62 0C 42 F5
0A39:CB 70 C6 0C 3C C0 4E 46 B0
0A41:ED 09 60 A6 34 D0 12 AC BA
0A49:74 68 6C 4F E2 66 0F 81 84
0A51:60 B9 01 04 C6 10 F2 30 F0
0A59:66 0E 0E 4E 70 06 F7 85 DB
0A61:42 DC 85 43 22 21 09 8C E6
0A69:84 22 26 6F CE 81 50 62 83
0A71:C9 8F 95 11 D2 08 25 4D 60
0A79:38 9F 01 1F D0 02 B0 47 DA
0A81:20 F4 81 87 4F 40 06 C0 D3
0A89:0F 49 CE 0F 31 9A 43 1D D9
0A91:05 1E DE 08 0C 72 12 F9 54
0A99:72 60 0C 85 F9 AD 26 FA A6
0AA1:E9 00 85 FA F7 00 6A C8 68
0AA9:BD D1 98 90 E8 44 68 16 6C
0AB1:F5 B0 03 67 BE 81 07 F4 C2
0AB9:07 A5 80 4B B0 F0 12 06 F2
0AC1:8C 97 07 A0 02 E0 6A 8B E0
0AC9:10 A8 06 6D 0E 29 81 73 37
0AD1:95 08 A6 0A C9 22 C1 05 87
0AD9:17 A2 37 D0 F1 84 70 00 98
0AE1:89 A9 30 1A C8 8A 91 06 66
0AE9:A4 0F 60 D0 60 66 21 10 1C
0AF1:14 02 42 60 A9 0A 60 20 35
0AF9:A5 1F 94 91 74 91 28 DE 6D
0B01:80 21 A2 E0 30 0E 44 2F 73
0B09:D1 F2 69 1A A5 13 85 0C 24
0B11:A4 02 B8 09 03 06 2D 81 AD
0B19:0C 60 E6 3A F2 03 D0 A5 B8
0B21:06 38 E5 19 17 20 46 85 E1
0B29:19 A5 07 E5 1A 18 85 2D DD
0B31:85 80 D1 81 8D 3D 08 53 41
0B39:8D 3E 08 A9 04 D1 F9 2B C7
```

```
0B41:80 20 10 8C E0 00 0C BE 48
0B49:4C 00 4A 25 AC 52 1A 10 14
0B51:70 08 11 72 08 F7 72 08 F7
0B59:40 01 2C A5 2C E0 ED 2C 9C
0B61:E6 EC 2D 36 71 45 C7 86 E5
0B69:94 15 60 92 16 3A 83 C0 A5
0B71:E0 13 41 60 14 1E 11 85 AB
0B79:D0 DE F0 A9 E0 12 90 05 DD
0B81:A9 09 88 F1 15 00 8A 0A A6
0B89:AA A9 A7 A9 B1 01 0D DD 78
0B91:C8 BD DC 82 80 D5 33 60 61
0B99:A2 07 D0 E8 F0 7E 86 F9 F3
0BA1:86 05 0E 5E 85 C2 83 AE D0
0BA9:85 CF 83 CD 84 8F 84 07 36
0BB1:88 D0 84 49 85 C9 83 C2 82
0BB9:87 E0 B8 E0 45 87 4E 85 5B
0BC1:20 7C 34 09 94 C1 A6 F8 10
0BC9:11 F7 90 73 1A C5 18 D0 98
0BD1:06 55 C5 17 B0 33 41 C0 FF
0BD9:82 F7 40 08 F8 72 08 32 8B
0BE1:3C 13 19 85 14 C7 0A A0 CA
0BE9:02 28 80 60 E1 A5 19 A5 9E
0BF1:96 19 A3 1A 4C 0E 83 A9 FA
0BF9:05 56 F3 0F 20 CC 8A 01 E1
0C01:40 08 A9 1A BA FF A2 7F AC
0C09:C0 FD 3D 95 00 2A 7F 10 B9
0C11:02 2D 32 2E A5 F9 8D 1B E9
0C19:08 A5 FA 8D 1F 08 6C 02 CA
0C21:A0 A5 2E F0 12 6E 3D B2 3F
0C29:80 04 F7 0C 01 6C 02 13 13
0C31:C6 2E D0 EA 5C 60 17 E1 75
0C39:30 18 4C 13 00 F1 18 69 8B
0C41:02 04 38 4D C0 8E 60 48 80
0C49:34 01 2C 90 DA 18 60 B6 F8
0C51:B9 C0 05 68 68 27 5D 08 40
0C59:90 84 D1 0F 2C 00 85 FD 70
0C61:A9 B2 7C 09 01 18 3C F0 EC
0C69:C2 A5 33 48 BE FD 43 CC 78
0C71:2F 0C 47 40 E8 27 04 A9 A6
0C79:8D D0 0F B6 50 48 31 22 02
0C81:8A C2 9D 85 40 4F A7 05 2F
0C89:61 50 70 9C 08 39 2A EA A2
0C91:28 11 1C A5 40 5E BE 09 E1
0C99:A5 FD 2E FC E2 7E 68 31 AC
0CA1:F8 8B A5 3F F0 C5 A9 BD 71
0CA9:65 18 02 02 8E 58 E6 E1 60
0CB1:C1 9D D8 C3 9D 08 23 1D D9
0CB9:E0 40 D0 15 44 C6 13 11
0CC1:AB 4C 48 84 E6 2E 98 4E 83
0CC9:02 41 9E DC E8 E0 09 D0 82
0CD1:F5 DC 84 40 58 18 F7 26 E9
0CD9:C2 10 A8 81 A9 A7 32 C8 9D
0CE1:0C A9 89 7B CD 0E B9 06 73
0CE9:D0 F6 88 04 60 92 69 A9 43
0CF1:4C 2C 27 52 08 BC CC 46 58
0CF9:0F F0 01 93 83 B0 53 C9 7E
0D01:65 2A A9 C0 85 2B 58 57 7A
0D09:41 18 13 C8 14 08 C5 14 19
0D11:F0 1C 95 10 18 69 02 85 4E
0D19:2A FA 71 2B A5 2B C5 39 86
0D21:58 6D 2A C5 38 90 DA C2 E0
0D29:4E 42 57 D1 1D 19 38 E9 AA
0D31:04 85 19 B0 02 C6 1A A5 E1
0D39:F7 58 4E 90 F8 43 90 13 41
0D41:43 A5 14 91 19 50 16 C4 FC
0D49:C8 B1 05 A3 30 3E 03 D0 60
0D51:07 A5 02 75 2E A9 60 50 19
0D59:B8 41 0E 36 84 A1 1D C7 F1
0D61:17 B0 12 12 20 6F CE 3B 2E
0D69:58 4E 00 14 3C 27 40 8E 12
```


PROGRAMS

```

0D71:92 94 26 E7 0E 2D 80 E1 45
0D79:81 3F D1 00 1E F0 0F C6 F7
0D81:3F CC 89 69 1C B2 A9 01 36
0D89:85 FD 4C 7A FF 07 01 BB F0
0D91:D5 68 DE 82 85 95 0D 28 79
0D99:AF 85 BC BA 60 E5 0F 30 19
0DA1:0B 80 C2 83 CA 68 84 17 0A
0DA9:F0 2E C9 3B F0 37 5A AE 7C
0DB1:26 C9 A3 F0 39 C9 A6 F0 04
0DB9:45 E2 DF 80 51 61 A5 3F CD
0DC1:D0 0D 2B 25 D9 72 72 80 3D
0DC9:72 86 5D 36 B3 F3 78 DD 0A
0DD1:61 04 D3 D8 D0 B8 D0 33 E3
0DD9:3C 6D 34 0F 19 63 86 A9 F1
0DE1:80 9E 0F 75 60 64 9C 59 44
0DE9:34 7B 80 0E 05 83 CD C9 89
0DF1:3A F0 C9 4C E1 85 C3 F5 06
0DF9:00 1C 09 88 4C 29 80 29 F5
0E01:20 ED 0F 22 B0 C3 2B D0 68
0E09:83 51 B8 09 C3 AB E0 A4 16
0E11:6C D0 13 1C 4C 1B 0B 71 12
0E19:E0 A9 D0 16 DB 0D C0 CD ED
0E21:86 A5 F8 B4 20 F7 38 0A AF
0E29:05 77 4C B3 91 41 90 01 1E
0E31:43 50 0C 02 22 D0 DF 82 3B
0E39:C2 0D 46 3F 84 1B 82 A2 EE
0E41:00 79 1F 18 10 07 C3 E3 29
0E49:A1 7A 01 D0 E9 A2 08 D4 70
0E51:A1 CA D0 FA 60 84 2A 90 B4
0E59:1C 54 84 21 30 AA 76 2E 82
0E61:C5 14 D0 04 E4 13 F0 10 25
0E69:76 78 86 4C 02 AB 61 00 F5
0E71:41 2A 73 3B 2B 70 22 0B 45
0E79:83 43 E4 17 85 3D AC 8A 3B
0E81:C8 C5 82 26 58 1E 9A A6 1F
0E89:7B 48 20 66 8E 89 39 46 33
0E91:8A 96 8C 5C 23 63 91 17 D0
0E99:4B 20 AF 95 07 56 5F 8D 90
0EA1:BD B1 60 67 0F 11 04 F0 41
0EA9:39 04 19 7D 07 A9 4C 0B E0
0EB1:CE CD E0 B4 19 8C 9C 0C 50
0EB9:33 41 F8 4C FA 86 60 A5 FB
0EC1:2A 30 19 2B 61 50 06 D1 FE
0EC9:5A A3 8B 48 20 D8 8B 68 D5
0ED1:34 B8 77 B7 8E 4C FF 8E D4
0ED9:60 69 E6 30 44 70 D0 0C F1
0EE1:33 C9 2C 3C B5 44 54 22 DC
0EE9:0C C1 82 61 01 8D F2 87 8E
0EF1:95 45 E8 E0 06 D7 60 28 CD
0EF9:C7 81 02 91 14 20 F1 25 DD
0F01:E2 86 B8 86 A1 0E 22 84 4F
0F09:E2 43 84 E8 78 8E 0A A9 44
0F11:11 41 EE 32 88 D0 0F EE 9D
0F19:49 2F 14 08 09 C9 72 05 A9 76
0F21:E0 8D 33 88 AD 0C A0 A2 83
0F29:37 86 01 AE 6B 88 58 47 05
0F31:9D 2E 20 E2 EC 43 0B 0F 75
0F39:84 8C 97 00 47 60 A2 7F 30
0F41:B5 00 9D 00 7F CA 10 F8 2E
0F49:4C 0A 80 86 73 B0 89 00 F9
0F51:01 37 A9 08 6B A2 00 BD 17
0F59:00 02 D1 AE 35 E4 0E 90 07
0F61:F5 4C 86 00 11 0F B0 C0 45
0F69:52 83 BA 31 A5 0C 08 BD 27
0F71:A6 88 85 09 01 43 10 88 83
0F79:F0 EA 34 C8 D0 F6 60 E5 E6
0F81:88 07 89 19 60 F1 00 89 BC
0F89:25 89 31 89 49 89 50 89 08
0F91:67 89 7B 89 90 89 A2 E9 A7
0F99:01 89 A9 89 B7 89 CD 89 71
0FA1:DE 89 EC 89 02 8A 10 8A AC
0FA9:24 8A 36 8A 49 DA 03 8A 32
0FB1:5D 8A 2C 16 F3 E0 28 41 BC
0FB9:07 1C 25 60 C8 68 0D 34 43
0FC1:7F D2 C2 63 57 41 B6 4B 5B
0FC9:C4 B3 08 D9 B0 2B 88 E2 FB
0FD1:42 C1 91 84 0B 9C 48 22 79
0FD9:6C 63 61 31 67 21 0E 79 9B
0FE1:E0 20 32 2E 2E 46 0C B8 FC
0FE9:00 DD 0D 46 9F 2C 50 01 D4
0FF1:6D 44 2E 8A 12 91 05 D0 F7
0FF9:C8 F8 D8 3A 20 20 B3 29 8B
1001:D6 53 55 50 50 30 73 C3 FE
1009:79 44 27 4B 45 9A 33 3C C7
1011:76 EA 46 55 4E 43 64 DB 69
1019:E0 D9 20 2D 93 1C 68 82 57
1021:2D 60 4C 6B CE 40 53 59 A7
1029:85 42 41 58 88 06 FC 22 C2
1031:81 4E 59 43 27 19 6C 90 0C
1039:4C 45 47 63 AB 54 39 E0 F1
1041:91 4D A9 20 55 AA C2 00 8F
1049:4E 45 58 F0 57 49 80 90 6D
1051:4F 55 28 46 4F 3E 00 54 9F
1059:59 50 90 0D 3C 0A 13 54 A1
1061:43 48 00 0D E5 28 8C 22 11
1069:41 54 49 C7 20 41 42 63 D2
1071:45 54 45 44 A6 55 E1 51 D5
1079:4D 4D 41 62 13 50 41 77 B5
1081:E9 00 48 8F D8 C0 00 E0 43
1089:DC A1 02 2E 43 4F 94 0F 33
1091:00 99 86 20 49 2F 4F D2 63
1099:52 52 84 A4 22 2A 2A 2A 8A
10A1:37 44 45 56 49 43 45 20 7E
10A9:4E 4F 54 61 3C 52 45 53 6E
10B1:45 4E 54 0D 00 4D 49 53 7E
10B9:06 B9 4E 47 20 45 51 55 97
10C1:41 4C 20 53 49 47 4E 00 D2
10C9:A2 FA 9A 48 A9 40 85 12 3D
10D1:82 77 0A 31 C1 6C 20 AC 12
10D9:8A A9 64 02 50 88 68 5A 26
10E1:2F 20 D4 A0 A9 3A 96 75 1F
10E9:C1 0A 49 D8 A5 49 61 82 BB
10F1:A5 3A C0 56 D3 80 4C 94 BE
10F9:82 C0 32 20 A5 10 75 1D 49
1101:11 0D A2 88 84 1F A2 DD 8F
1109:42 2A 40 48 2A DD 51 07 D5
1111:60 8D 2A 38 FD 23 48 03 9F
1119:2A A5 2B FD 24 C5 2B C8 56
1121:D0 E0 C7 41 85 C0 53 A4 6B
1129:20 70 9E 98 09 30 4C A1 18
1131:E6 20 3A 52 C6 CC 9B D2 AE
1139:38 4F 3A CC 17 10 0A A6 52
1141:34 20 13 20 2D 8B 4C 19 33
1149:8B D8 0A 06 30 56 00 A9 8D
1151:22 EB A5 C8 C4 0E 90 DF 20
1159:0D 54 1A 98 89 C2 38 E9 95
1161:7F AA 33 B6 FF CA F0 08 D4
1169:3C C8 10 FA 30 F5 C8 B9 32
1171:9E A0 30 06 E3 07 4C 40 85
1179:8B 29 7F ED 85 22 56 80 5C
1181:81 82 83 85 86 87 88 89 BA
1189:8B 8C 8D 8E 8F 97 99 9E DB
1191:A1 B4 B6 B9 BB C2 C3 C4 59
1199:C5 C6 C7 CA C1 C7 CA 0B C3
11A1:43 12 88 55 91 12 53 51 1D
11A9:88 A4 27 88 C0 40 69 9A 1B
11B1:20 E8 32 F0 27 73 CE 0A 22
11B9:86 12 86 21 86 12 AB A2 FC
11C1:86 0F A6 27 FC 3A 84 01 0D
11C9:A2 00 02 0F C6 A9 86 27 7F
11D1:A6 23 82 84 82 A0 48 73 43
11D9:90 AD 82 41 00 3A 4C 80 16
11E1:8B 46 90 A1 96 20 0A 19 E9
11E9:04 19 27 A3 8B 34 06 84 31
11F1:98 40 91 F7 E6 F7 44 C2 84
11F9:08 F8 A5 F8 C9 C0 36 AF 10
1201:F7 01 B0 03 A4 0F 41 23 AE
1209:0D 89 01 18 B6 01 C3 1E 17
1211:AC 03 91 FB A2 18 1A 10 F7
1219:80 8A 8D 1B AA F0 1D 23 F9
1221:C8 10 31 85 11 A8 20 0F A6
1229:15 88 B5 25 0A 9D 28 15 2F
1231:E8 E0 50 B0 0B 90 EF 86 18
1239:0E E2 3B 45 34 AC 80 05 33
1241:20 8C 8E 2E A1 90 81 28 07
1249:40 A4 32 C4 0E B0 1D 77 2E
1251:02 8D 6C C8 48 27 04 45 20
1259:34 85 34 61 55 01 04 B3 FF
1261:F0 E6 84 32 84 16 10 03 0B
1269:62 19 2C F3 E2 90 88 C3 F7
1271:06 1C 86 70 00 8A 84 26 D0
1279:A4 31 72 C7 85 32 C8 84 12
1281:31 A8 B9 FF 01 85 33 A4 C8
1289:26 60 85 35 30 5D 09 10 F5
1291:82 80 09 C5 35 F0 0A 8D A3
1299:CB F7 19 03 04 45 34 D0 63
12A1:F2 E6 31 BB 72 08 06 5B F5
12A9:61 41 68 18 36 2C 3A 9A CE
12B1:64 30 90 01 60 40 22 03 81
12B9:00 85 36 6E 05 5E 60 A3 F2
12C1:33 C9 AB 13 E6 36 A2 44 32
12C9:82 32 85 37 97 95 2F 29 7A
12D1:CF 85 16 A5 1D 15 21 15 F0
12D9:20 0A 26 50 15 00 4C A5 41
12E1:15 65 07 06 13 26 14 A5 2A
12E9:13 65 16 D3 70 10 40 14 49
12F1:6E 6A CE F0 0B A5 32 C5 E9
12F9:37 32 F8 00 90 A5 36 F0 DD
1301:0D D3 38 E5 13 C0 10 61 25
1309:E5 14 96 0E 60 85 6C 10 DC
1311:23 20 0A 30 94 00 90 09 E3
1319:85 41 E8 18 84 D0 ED C9 FD
1321:25 32 1A 18 2A C9 24 D0 BC
1329:09 72 21 20 04 D4 F0 1D 09
1331:C9 28 D0 19 A5 1F 1B 5E 30
1339:50 28 D0 0B 54 0F 01 8E CB
1341:14 1E F0 04 A9 10 85 1E 0E
1349:E0 10 E2 54 A2 0F 86 1D 01
1351:28 3E E8 20 E1 FF 6B 5E 7E
1359:84 88 94 F6 C0 F0 40 29 59
1361:0F 85 20 C5 1D D0 28 E3 31
1369:08 29 F0 C5 1E D0 20 4C 19
1371:90 0C E3 3C 28 A2 00 BC AB
1379:76 07 DD D1 98 D0 07 E8 74
1381:E4 1D 90 F3 C7 63 A5 20 E9
1389:A8 79 D0 02 E6 23 4C A1 9A
1391:8D 4D 0B EC 81 A0 3F 6D 7C
1399:6C B0 25 2B AF 19 C9 C4 B3
13A1:F0 15 C9 C7 90 04 C9 CB 2F
13A9:90 0D 41 73 6B B0 11 4A 45
13B1:19 D0 08 72 1D E6 3F 4C 10
13B9:86 94 E8 41 43 32 C0 21 FE
13C1:26 4D C9 AE B8 2E 20 87 B8
13C9:99 40 60 20 3C E9 2A 3E F6
13D1:78 13 0F F6 54 09 02 D1 E6
13D9:AF 18 50 6E 82 04 1E 91 C0
13E1:40 C8 09 A6 8E 4A 4E D0 F0
13E9:7E 53 90 BD 7F 35 0E BE BD
13F1:80 2C 8E 60 3D 09 4B 09 E8
13F9:59 09 84 09 E4 0D D3 09 3C

```


1401:E0	09	25	0A	1B	0A	31	0A	8E	1631:0A	42	0A	38	0A	4C	0A	54	A1	1861:00	F0	05	D6	F8	26	EC	67	7D
1409:45	0A	3B	0A	4F	0A	D2	8D	34	1639:DD	0C	B4	D0	22	20	59	10	4F	1869:30	25	08	C3	25	0E	C3	02	23
1411:20	00	11	52	C9	AB	F0	01	70	1641:5D	91	A5	D8	D6	20	A1	A2	DF	1871:05	04	5E	10	C0	05	79	83	82
1419:25	54	01	2C	A1	24	01	3D	A8	1649:49	C8	13	21	A9	A0	A8	08	E9	1879:11	38	30	11	20	88	0B	A5	36
1421:86	3E	E7	9A	30	E3	9A	E1	EA	1651:A5	14	AA	94	4C	BD	92	C9	3C	1881:03	C5	05	56	AF	02	C5	04	BF
1429:71	94	E1	CB	83	99	E1	24	92	1659:80	90	06	20	A7	8F	4C	77	38	1889:60	38	60	AE	30	EF	A9	01	84
1431:30	62	E6	18	69	01	A6	3E	43	1661:91	20	C3	8C	B0	AA	37	00	3E	1891:18	52	03	FF	31	D8	E0	02	73
1439:B5	2C	84	84	8C	D0	27	D7	ED	1669:3C	34	20	47	8D	A5	1E	C9	42	1899:85	03	60	2F	13	E6	EA	D0	27
1441:07	20	F1	A5	3D	F0	B6	34	DC	1671:20	F0	05	A9	10	4C	70	8A	42	18A1:EF	AB	98	E7	90	E5	B0	DC	A0
1449:DC	A0	C9	F0	C9	20	3A	2B	BE	1679:A9	AD	48	EF	20	03	8C	E6	FA	18A9:37	21	B0	DB	90	D2	CE	89	88
1451:D8	20	B9	B2	14	5B	8F	20	9D	1681:13	6B	85	14	D0	B7	BB	8B	6C	18B1:D1	D0	C8	F9	12	E2	90	C7	BC
1459:CD	30	22	0E	F3	0D	A9	BD	7E	1689:A2	A1	04	A9	9A	34	91	6A	BD	18B9:B0	BE	20	01	09	20	ED	09	B3
1461:2F	8F	22	40	C0	B1	94	01	44	1691:A5	33	F0	0D	C9	22	F0	09	0E	18C1:F0	B6	90	B4	B0	B9	A5	05	32
1469:20	2E	72	20	9B	8E	36	20	21	1699:9D	E2	37	70	E8	D0	EF	86	2C	18C9:85	17	4C	3D	09	A5	17	30	1D
1471:DD	8C	6A	A1	01	63	35	0A	1D	16A1:1D	20	4D	8C	A5	06	D0	F1	AF	18D1:03	4C	4F	0A	4C	45	20	FC	D5
1479:E8	B4	77	E4	F4	AA	CA	61	C9	16A9:1A	A5	07	85	23	A4	3D	22	CD	18D9:00	2C	A2	BB	05	08	C9	E0	E1
1481:51	1B	40	00	F0	0E	C9	10	84	16B1:C9	30	D0	21	AC	41	C5	1D	0E	18E1:02	01	19	09	85	0F	77	79	D7
1489:D0	BC	11	02	1C	AC	41	C3	65	16B9:D0	18	49	90	13	C8	B1	87	2D	18E9:98	02	03	34	00	85	0E	C4	81
1491:4C	6E	8F	A2	00	DD	54	8B	43	16C1:14	A0	FF	C8	C4	1D	F0	18	41	18F1:F0	F0	3E	B9	02	DC	C9	2D	8D
1499:F0	0A	E8	E0	1F	90	F6	A3	B4	16C9:B1	13	D9	D1	98	F0	F4	A9	08	18F9:8A	58	E6	0E	D0	30	C9	30	4E
14A1:9F	03	E0	12	AD	60	0E	C0	63	16D1:30	29	F0	18	69	04	65	22	0C	1901:90	2F	C9	3A	B0	2B	E9	2F	59
14A9:C8	02	8A	E9	12	0A	AA	BD	72	16D9:85	22	90	CC	6E	23	D0	C8	5E	1909:9D	C7	B5	03	85	0C	61	05	07
14B1:D2	82	0F	BD	D1	8F	48	4C	4B	16E1:60	20	5A	91	A6	FC	A5	29	48	1911:08	0A	26	54	05	02	F2	00	E9
14B9:74	91	00	0F	80	02	08	ED	7B	16E9:D0	0C	BD	DE	E1	85	5F	D8	E0	1919:8C	75	03	95	03	16	02	36	53
14C1:8F	EA	8F	1B	90	B7	90	3B	CF	16F1:DF	59	C8	BD	E0	92	20	2C	17	1921:03	B5	02	65	0D	95	02	90	2C
14C9:90	59	90	7F	90	D9	90	FA	A2	16F9:BD	E1	92	4C	E3	8B	A8	08	3B	1929:02	F6	03	C8	D0	BE	A5	0E	E1
14D1:A0	0E	01	59	00	1D	07	5D	63	1701:C0	08	06	40	9E	0C	10	0C	A7	1931:F0	03	32	78	9E	60	01	00	E2
14D9:0C	CE	42	E1	70	BD	12	90	51	1709:16	0C	2B	0C	31	0C	0B	08	43	1939:0A	00	64	00	E8	86	9F	27	C4
14E1:18	69	03	F7	50	13	CE	50	0D	1711:0A	00	9E	32	30	36	31	00	F7	1941:20	E6	0A	A6	28	66	A1	5C	63
14E9:86	08	4D	08	68	08	76	08	D9	1719:00	00	C0	FD	DA	0E	D4	8D	85	1949:B4	A9	7E	2C	A9	7F	8D	47	80
14F1:5A	08	1E	F7	08	44	4A	A0	13	1721:0F	90	04	80	8D	12	D4	68	4A	1951:0B	38	E9	76	80	50	0B	F6	76
14F9:63	70	E3	6B	23	2E	23	2C	67	1729:0F	30	C4	33	AE	02	91	33	8A	1959:1C	BD	0C	F1	03	84	82	10	E8
1501:4C	D5	4B	09	3A	2E	A6	28	C0	1731:00	98	40	A5	03	C9	7E	F3	18	1961:42	2C	2D	45	B1	D8	3F	6C	95
1509:BD	54	90	7C	2D	01	55	90	A9	1739:18	C9	00	90	E9	20	E2	0B	8F	1969:80	A2	05	0D	EC	DD	A4	DD	B8
1511:D6	47	69	0A	6C	17	00	85	8B	1741:4C	FF	60	F1	CC	38	F5	41	35	1971:42	2A	D0	05	C8	DD	40	0E	06
1519:29	61	94	1B	5B	28	39	76	D8	1749:7C	F5	EC	A4	05	02	A7	10	25	1979:90	13	88	A9	FD	CF	32	00	F7
1521:89	C8	2A	92	84	2D	7B	2A	AA	1751:1E	BA	A0	D1	B0	A1	C3	14	16	1981:22	03	FD	D0	0A	85	03	E6	A5
1529:09	72	6E	EA	F8	02	6C	40	D9	1759:88	84	8C	E0	0B	D3	1F	95	08	1989:00	D0	DF	E3	3F	45	42	C9	0E
1531:04	18	2D	40	8D	D0	10	00	DC	1761:E2	61	83	20	02	AD	1B	D4	9D	1991:0D	D0	04	A5	0C	6D	A5	DD	41
1539:E4	3C	7E	E4	50	14	AA	00	2B	1769:33	A4	18	DD	40	D0	05	B5	40	1999:09	30	99	A5	7D	E6	0C	CA	54
1541:5D	87	01	4A	08	85	72	34	30	1771:02	D0	01	60	B5	03	30	09	1E	19A1:5D	BF	84	00	60	46	3C	32	C9
1549:08	18	65	29	32	43	EA	A6	D7	1779:40	6E	99	A9	01	B5	A9	FF	63	19A9:28	1E	14	0A	A2	06	A5	D3	E6
1551:39	81	46	F8	07	0A	A9	E6	6B	1781:59	65	03	8C	8B	B2	78	2E	25	19B1:DD	85	74	05	90	06	CA	F6	3C
1559:5C	C2	E9	3D	7F	0A	39	7F	89	1789:2E	7E	7C	2E	32	80	E0	22	58	19B9:4C	B3	0B	BD	52	0B	85	02	08
1561:19	20	1F	1E	31	19	1C	04	10	1791:8D	CA	08	7F	02	FE	08	B1	01	19C1:20	2E	01	AE	D3	AA	E0	00	A5
1569:39	1C	75	94	64	A4	54	78	06	1799:06	BB	13	7F	A1	84	09	60	A5	19C9:F0	BA	03	25	CA	10	F4	79	CF
1571:C8	6C	88	24	C2	02	5C	4C	91	17A1:86	06	84	07	28	78	C4	AB	ED	19D1:85	5F	96	59	5C	85	E4	FF	C9
1579:44	C8	AE	85	72	A5	32	85	3A	17A9:CC	08	88	29	91	06	A0	26	EF	19D9:F0	05	8D	A1	72	E6	08	10	E0
1581:41	60	A9	02	85	28	A9	40	C5	17B1:F4	98	A8	A7	A4	08	2A	0A	B3	19E1:B8	F0	0A	A9	3F	9A	12	A9	BA
1589:E6	AA	80	0D	05	20	C4	0D	32	17B9:69	5E	8A	A6	E8	C8	D0	F2	EE	19E9:20	54	56	85	20	CF	FF	C9	6E
1591:23	68	20	05	A9	8E	20	18	9B	17C1:84	C0	92	03	E2	0A	78	0C	21	19F1:00	F0	06	06	F0	8B	42	A9	EB
1599:58	8F	8C	4A	4A	41	85	32	9E	17C9:61	06	08	1C	09	C9	A2	FF	A1	19F9:C0	A9	0D	4C	30	20	6C	02	3A
15A1:A5	72	85	33	19	FC	0D	A9	9F	17D1:E8	30	EE	13	E4	09	F0	0F	CB	1A01:00	A4	12	7D	81	06	F0	07	85
15A9:FD	20	E0	11	14	E0	D0	20	ED	17D9:C0	26	7E	DD	90	F0	EF	01	C8	1A09:9D	7B	C8	91	18	F5	86	08	CA
15B1:D0	08	40	A9	A5	02	4D	8D	46	17E1:7D	BD	00	7F	85	60	35	F8	47	1A11:23	D0	0E	A8	06	48	C8	B1	EB
15B9:48	1A	A9	2C	A0	1C	14	A9	CD	17E9:00	8D	14	95	03	A5	08	F0	07	1A19:10	CA	07	38	10	A0	00	84	F3
15C1:12	F2	A8	28	2C	A9	F3	24	5D	17F1:03	AD	00	7E	95	02	E4	0D	80	1A21:12	56	6E	85	10	41	C9	09	3C
15C9:A4	8C	88	A3	13	E1	4C	68	D5	17F9:04	D2	02	CB	10	65	03	55	4D	1A29:11	4C	DD	0B	A2	00	E4	08	4C
15D1:8A	8A	A4	62	20	48	94	0A	F3	1801:51	38	EE	9C	15	86	C5	59	37	1A31:F0	09	BD	00	7E	9A	57	10	F4
15D9:6B	5D	C9	AC	90	09	15	01	E8	1809:71	BB	F1	0B	21	3E	85	50	2D	1A39:E8	D0	F3	0E	A1	85	14	A5	66
15E1:CD	07	46	B3	C9	AA	90	4C	1F	1811:A1	93	90	0D	F5	18	65	40	F4	1A41:03	85	15	60	E0	04	85	02	25
15E9:CC	48	E9	A9	F0	07	68	89	62	1819:47	65	3B	47	45	18	66	93	C8	1A49:70	C0							


```

1A91:28 C3 61 4C C0 0B 01 60 50
1A99:20 BA FF F6 64 A5 C3 2A 67
1AA1:01 C4 48 A5 2B 94 A5 2C 0E
1AA9:A0 05 A9 C3 A6 2D A4 2E 42
1AB1:20 D8 E1 3B 85 C4 68 85 B1
1AB9:C3 6A B7 FF 8D 35 97 60 32
1AC1:20 D0 20 5C 32 41 36 01 08
1AC9:AC 33 97 20 BC 18 31 97 56
1AD1:A6 2B A4 2C 20 D5 FF 86 5A
1AD9:2D 84 2E 4C C6 97 AD 34 74
1AE1:97 A2 D5 A0 97 4C BD FF B8
1AE9:A9 00 20 42 07 9F 68 68 0B
1AF1:4C 59 A6 A9 C0 4C 90 FF 6A
1AF9:00 FF FF C0 7C 7F FE 4C 66
1B01:53 44 AF B2 E5 43 E9 D2 F5
1B09:83 64 C9 43 E2 C4 03 E1 99
1B11:0E 82 8D 00 BA BA BA AD 84
1B19:0C D0 20 A2 00 BD E9 9B 1E
1B21:9D EC CF E8 E0 0F D0 F5 C4
1B29:81 EE 12 58 0F E9 EE 0E AF
1B31:5D AC EC 8D 26 46 CF 8D 2F
1B39:27 60 79 06 22 93 50 23 CD
1B41:53 14 20 D0 8D 21 D0 B5 7F
1B49:35 86 02 15 79 37 A9 80 C9
1B51:85 38 EB C9 4C F0 32 C9 C6
1B59:46 F0 09 C9 45 F0 0F C9 82
1B61:55 F0 1E D3 74 C6 67 82 8F
1B69:A9 8C 8D 86 8A 20 B2 9A 86
1B71:AD 4C 4E D0 03 44 0F 45 F4
1B79:77 71 59 88 50 3D 01 8D 82
1B81:82 6F 9B 4C 8D 9A 20 73 97
1B89:0C DE AD 21 3D CD 42 F0 DB
1B91:04 B0 0E 90 0F AD 04 9C 94
1B99:CD 45 66 F0 0A B0 02 90 8B
1BA1:D4 02 42 0A 0A 01 01 E1 E3
1BA9:01 33 10 AD 11 9C F0 45 2C
1BB1:13 4E 19 DA F0 08 19 CA 7A
1BB9:09 21 FF 10 8E 4F 1C 8E 36
1BC1:02 79 38 27 1A 8E 0A A9 99
1BC9:03 8D 0A 4F 1B 08 8D 0B 3A
1BD1:9C 20 91 9A C9 01 E2 84 D6
1BD9:33 0D 61 85 87 8D 82 08 F0
1BE1:C2 8D 83 08 38 A5 2D E9 6A
1BE9:62 2D 90 8D 08 A5 2E E9 A4
1BF1:07 14 4A 8D 35 4C 40 37 65
1BF9:E2 03 91 05 47 B7 16 43 6D
1C01:97 11 A9 15 97 57 E1 01 AE
1C09:8D 14 97 97 E2 3B 9B AD 62
1C11:12 59 00 08 AD 13 9C 8D A9
1C19:78 AD 20 39 11 AD 55 4E C8
1C21:0C A9 42 E7 73 60 AD 0C 15
1C29:1F 2D AD 0D 81 C1 2E 78 AA
1C31:A9 35 A0 08 B8 C0 85 05 F8
1C39:A9 08 08 86 A9 A0 D0 0D 30
1C41:18 B1 05 91 03 38 CE 0F 51
1C49:01 27 1B CE 10 9C D0 16 C6
1C51:A9 37 85 01 58 A9 1E 85 18
1C59:03 A9 9C 85 04 20 F5 9B 91
1C61:A9 00 8D 11 9C 60 C0 C1 DA
1C69:27 0E E6 04 E6 06 A5 06 76
1C71:C9 C8 D0 04 A9 E0 85 06 FC
1C79:4C B0 9B 08 48 AD 8D 02 0E
1C81:D0 FB 68 28 4C CA F1 A0 C2
1C89:00 A9 0D 10 E1 0D B1 03 78
1C91:F0 07 20 D2 FF C8 4C FC ED
1C99:9B 60 29 23 28 23 9A 3E 50
1CA1:90 11 00 00 40 08 53 41 70
1CA9:56 5D 11 82 20 52 88 48 52
1CB1:86 0B 2C 00 4F 20 0E 59 E5
1CB9:4E 47 10 49 12 53 35 C7 91

```

```

1CC1:42 85 07 20 57 48 9F 5D D7
1CC9:3F 00 12 4C 4F 43 4B 01 C7
1CD1:87 21 21 00 93 57 45 4C 0B
1CD9:02 DE 52 68 54 4F 50 2E 4A
1CE1:2E 11 11 12 80 06 DD 49 D9
1CE9:92 75 0D 22 1E 19 3C 20 7A
1CF1:45 53 40 2E 42 41 53 49 93
1CF9:43 20 43 C0 3B 50 49 4C 4A
1D01:45 70 46 4F 52 20 54 48 BB
1D09:45 20 36 34 C0 B8 55 50 DB
1D11:44 41 54 45 09 5F 46 52 41
1D19:4F 4D 20 53 50 00 67 4E 27
1D21:54 20 49 49 20 42 59 20 28
1D29:46 41 52 49 44 20 41 48 23
1D31:4D 41 44 0D 0D 00 70 C1 C6
1D39:0F 64 04 EA EA EA 00 00 46

```

Farid Ahmad programs his 64 in Islamabad, Pakistan.

PAD DESIGN

By Robert C. Marcus

Impedance matching is an important concept in radio and electronics work. To get the maximum power from your ham radio to your antenna, for example, you'll have to match the impedance of your antenna to the impedance of your transmitter.

If you want a filter to pass a desired band of frequencies, you'll have to make sure the input and output impedances of the filter match those of your input and output circuits.

If you want to get the most power out of a stereo speaker, then the speaker impedance must match the output impedance of your amplifier.

If you're a ham radio operator or an electronics hobbyist, these are just a few of the times when you might find it necessary to design an attenuator network, or what's generally referred to as a pad.

Whether you want to insert additional attenuation or merely match impedances, the math that's involved can be intimidating. That's where Pad Design can help. It does all the number crunching necessary to calculate the resistive elements necessary to meet the design requirements that you specify.

The program is set up to deal with unbalanced T and Pi networks, but it can also deal with balanced H and O networks. The minimum-loss pad for impedance matching is included as well.

In many applications, it's possible to use commercial standard-value resistors. Their low cost and availability are trade-offs for a slight compromise in the attenuation and impedances being matched.

The advantage of using Pad Design is that it lets you try out various standard values to produce a network that is closest to the design criteria.

Pad Design is written entirely in BASIC. To help avoid typing errors, enter it with The Automatic Proofreader. See "Typing Aids" elsewhere in this section.

Using the Program

When the program is run, you can choose the type of network to be used: T, H, Pi, O, or MIN-LOSS. Pad Design then prompts for the input and output impedances that will be used, and if the network is a T/H or Pi/O, the insertion loss in decibels is asked for. The program then displays the network and gives the values required for the resistive arms.

At this point you may select to end the program, return to the menu, or changes resistors to obtain the best choice of standard values.

If you decide to make changes to the resistors, you'll be prompted for new values. The program limits the deviation of value to plus or minus 20 percent of the actual design value; this is more than adequate when you are using standard resistors whose values may vary by 10 percent. For more precise values, 5- or 1-percent resistors would be items of choice. If the original design value is to be retained for a particular resistor, just press Return.

After you make your changes, the network is displayed again. This time it lists two sets of impedances. The first gives the value of Z_I, which results from the new R values when Z_O terminates in the originally specified impedance. The second value gives Z_O as it appears when Z_I terminates in its specified impedance. The insertion loss—calculated with the existing changes—is displayed as well.

Balanced Networks

When dealing with balanced networks, remember that the value of the series arm or arms is halved. Since the values given and entered are on a single-unit basis, you'll have to multiply by two when choosing standard values.

Decimal Places

The calculated values for resistors and impedances are rounded off at two dec-

imal places for values below 1 ohm and one decimal place for values between 1 and 1000 ohms. Attenuation values are rounded off at two decimal places.

Should you wish to set your own values for resistors and impedances, you can alter the multiplier. The multiplier is the function FNPM(x), which is defined in line 110. It is, in effect, powers of 10 used with the INT() function to set the number of decimal places, such as $INT(FNPM(n)*n+.5)/FNPM(n)$ where n is a floating point number.

The function can be interpreted as $FNPM(x)=10^{(BASE-(x<LL+(x>HL)))}$.

Base is the number of places that are to be rounded: 1 sets one decimal place, 2 sets two decimal places, and so on. LL is the lower limit. Below this value the number of places increases by one over the base number. HL is the high limit at which the number of places decreases by one from the base.

To sum up our application, there's one decimal place (base) between 1 (LL) and 1000 (HL), there are two decimal places below 1, and there are no decimal places for 1000 and over.

Programmers using this method for rounding off floating point numbers should note that making the base a negative number produces rounding to tenths, hundredths, and so on.

PAD DESIGN

```
BA 10 REM COPYRIGHT 1992 COMPU
TE PUBLICATIONS INTL LTD
- ALL RIGHTS RESERVED
KQ 100 REM ***{2 SPACES}INITIA
LIZE{2 SPACES}***
PB 110 HD$="{RVS}{10 SPACES}PA
D DESIGN{9 SPACES}":DEF
FNPM(X)=10^(1-(X<1)+(X>
1000))
CM 120 CC$(1)="":CC$(2)="
{CYN}RESULTS WITH NEW '
R' VALUES{7}"
DP 130 LOS(1)="INSERTION LOSS"
:LOS(2)="MINIMUM-LOSS"
FE 140 AN$(1)="T OR H":AN$(2)
="↑ OR O":AN$(3)="MIN-
LOSS"
DE 150 POKE783,PEEK(783)AND254
:POKE53281,0:POKE53280,
0
JE 160 REM ***{2 SPACES}MAIN M
ENU{2 SPACES}***
AC 170 PRINT"{CLR}";:FORI=1TO4
0:PRINT"{RVS} ";:NEXT
```

```
RS 180 PRINT"{HOME}"TAB(5)HD$
KJ 190 RO=3:CO=5:GOSUB1150:PRI
NT"SELECT:":CO=10:FORI=
1TO3
PQ 200 RO=RO+2:GOSUB1150:PRINT
I" - "AN$(I):NEXT
JG 210 INPUT"{DOWN}CHOICE:":PS
:IFPS<"1"ORPS>"3"THEN21
0
JG 220 PH=1
PC 230 INPUT"{DOWN}ENTER INPUT
{2 SPACES}IMPEDANCE":ZI
:IFZI=0 THENGOSUB1260:G
OTO230
GH 240 INPUT"ENTER OUTPUT IMPE
DANCE":ZO:IFZO=0THENGOS
UB1260:GOTO240
HM 250 IFPS="{3"ANDZO=ZITHENPRI
NT"{DOWN}{RVS}MIN-LOSS
{SPACE}IS TO MATCH Z'S"
:GOTO1210
KS 260 IFPS="{3"THEN300
SS 270 INPUT"ENTER ATTENUATION
IN DB.":AT
BQ 280 IFAT=0THEN1250
BP 290 N=EXP(LOG(10)*AT/10)
MJ 300 ONVAL(PS)GOTO310,440,57
0
RB 310 REM ***{2 SPACES}CALC F
OR T OR H{3 SPACES}***
FQ 320 R3=2*SQR(ZI*ZO/N)/(N-1)
FS 330 R1=ZI*(N+1)/(N-1)-R3:R1
=INT(FNPM(R1)*R1+.5)/FN
PM(R1)
DA 340 R2=ZO*(N+1)/(N-1)-R3:R2
=INT(FNPM(R2)*R2+.5)/FN
PM(R2)
KQ 350 R3=INT(FNPM(R3)*R3+.5)/
FNPM(R3)
FK 360 IFSGN(R2)=-1THEN1230
JP 370 RA=R1:RB=R2:RC=R3
HA 380 GOTO690
HM 390 ZA=(ZO+R2)*R3/(ZO+R2+R3
)+R1:ZA=INT(FNPM(ZA)*ZA
+.5)/FNPM(ZA)
SF 400 ZB=(ZI+R1)*R3/(ZI+R1+R3
)+R2:ZB=INT(FNPM(ZB)*ZB
+.5)/FNPM(ZB)
KX 410 VL=1/((ZO+R2)*R3/((ZO+R
2+R3)*ZA)*ZO/(ZO+R2))
GJ 420 AT=INT(100*(20*LOG(VL)/
LOG(10)-10*LOG(ZA/ZO)/L
OG(10))+.5)/100
SD 430 GOTO690
HD 440 REM ***{2 SPACES}CALC F
OR ' OR O{3 SPACES}***
GR 450 R3=(N-1)/2*SQR(ZI*ZO/N)
PB 460 R1=1/(1/ZI*(N+1)/(N-1)-
(1/R3)):R1=INT(FNPM(R1)
*R1+.5)/FNPM(R1)
AX 470 R2=1/(1/ZO*(N+1)/(N-1)-
(1/R3)):R2=INT(FNPM(R2)
*R2+.5)/FNPM(R2)
CE 480 R3=INT(FNPM(R3)*R3+.5)/
FNPM(R3)
AB 490 IFSGN(R1)=-1THEN1230
RD 500 RA=R1:RB=R2:RC=R3
DH 510 GOTO870
```

```
XX 520 OZ=ZO*R2/(ZO+R2):IZ=ZI*
R1/(ZI+R1)
XP 530 ZA=(OZ+R3)*R1/(OZ+R1+R3
):ZA=INT(FNPM(ZA)*ZA+.5
)/FNPM(ZA)
BS 540 ZB=(IZ+R3)*R2/(IZ+R2+R3
):ZB=INT(FNPM(ZB)*ZB+.5
)/FNPM(ZB)
FX 550 AT=INT(100*(20*LOG((OZ+
R3)/OZ)/LOG(10)-10*LOG(
ZA/ZO)/LOG(10))+.5)/100
HP 560 GOTO870
XD 570 REM ***{2 SPACES}CALC F
OR MIN-LOSS{2 SPACES}**
*
EP 580 IFZO>ZITHENZ=ZI:ZI=ZO:Z
O=Z
PA 590 R3=ZO/SQR(1-(ZO/ZI)):R3
=INT(FNPM(R3)*R3+.5)/FN
PM(R3)
DM 600 R1=ZI*SQR(1-(ZO/ZI)):R1
=INT(FNPM(R1)*R1+.5)/FN
PM(R1)
XG 610 ZA=ZI
MP 620 RA=R1:RB=R2:RC=R3
XS 630 GOTO660
GH 640 ZA=ZO*R3/(ZO+R3)+R1:ZA=
INT(FNPM(ZA)*ZA+.5)/FN
PM(ZA)
BD 650 ZB=(ZI+R1)*R3/(ZI+R1+R3
):ZB=INT(FNPM(ZB)*ZB+.5
)/FNPM(ZB)
MX 660 VL=ZA/(ZO*R3/(ZO+R3))
XJ 670 AT=INT(100*(20*LOG(VL)/
LOG(10)-10*LOG(ZA/ZO)/L
OG(10))+.5)/100
XR 680 GOTO1000
MA 690 REM ***{2 SPACES}DISPLA
Y T NETWORK{2 SPACES}**
*
HD 700 PRINT"{CLR}"HD$
{2 SPACES}"AN$(1)
MK 710 PRINTTAB(5)CC$(PH)
SQ 720 PRINTTAB(49)"CCCC{RVS}
{WHT}{3 SPACES}{7}{OFF}
CC{R}CC{RVS}{WHT}
{3 SPACES}{7}{OFF}CCC"
PF 730 PRINTTAB(10)"{8 SPACES}
B"
MR 740 PRINTTAB(10)"{3 SPACES}
R1{3 SPACES}{RVS}{WHT}
{SPACE}{7}{OFF}
{4 SPACES}R2"
ME 750 PRINTTAB(10)"{8 SPACES}
{RVS}{WHT}{7}{OFF} R3"
HP 760 PRINTTAB(10)"{8 SPACES}
{RVS}{WHT}{7}{OFF}"
KP 770 PRINTTAB(10)"{3 SPACES}
{WHT}{3 @}{7}{2 SPACES}
B{2 SPACES}{WHT}{3 @}
{7}"
GS 780 PRINTTAB(9)"CCCCCCCC
{E}CCCCCCCC"
XJ 790 PRINTTAB(10)"{3 SPACES}
{WHT}{T}{T}{T}{7}
{5 SPACES}{WHT}{T}{T}
{7}"
SM 800 PRINT"{3 DOWN}↑ NOTE: I
```



```

F BALANCED H THEN R1 &
{SPACE}R2"
AQ 810 PRINT "{8 SPACES}ARE HAL
VED."
AJ 820 PRINT "{DOWN}R1="R1TAB(1
5)"R2="R2TAB(30)"R3="R3
XQ 830 RO=5:CO=3:GOSUB1150:IFP
H=1THENPRINT"ZI"ZITAB(3
0)"ZO"ZO:GOTO850
FB 840 PRINT"ZI{CYN}"ZATAB(30)
"Z7"ZO"ZO:RO=7:GOSUB115
0:PRINT "{2 SPACES}"ZITA
B(30)"{2 SPACES}{CYN}"Z
B"Z7"
XF 850 RO=11:CO=10:GOSUB1150:P
RINTLOS (INT (VAL (P$)/2+.
5));AT" DB"
HR 860 GOTO1160
KA 870 REM ***{2 SPACES}DISPLA
Y ↑ NETWORK{2 SPACES}**
*
CB 880 PRINT "{CLR}"HDS"
{2 SPACES}"ANS(2)
SF 890 PRINTTAB(5)CC$(PH)
KB 900 PRINTTAB(50)"CCCCC{R}CC
{RVS}{WHT}{3 SPACES}
{OFF}{7}CC{R}CCCC"
KH 910 PRINTTAB(10)"{5 SPACES}
B{7 SPACES}B"
XH 920 PRINTTAB(10)"{5 SPACES}
{RVS}{WHT} {OFF}{7}
{3 SPACES}R3{2 SPACES}
{RVS}{WHT} {OFF}{7}"
BX 930 PRINTTAB(10)"{5 SPACES}
{RVS}{WHT} {OFF}{7} R1
{4 SPACES}{RVS}{WHT}
{OFF}{7} R2"
AG 940 PRINTTAB(10)"{5 SPACES}
{RVS}{WHT} {OFF}{7}
{7 SPACES}{RVS}{WHT}
{OFF}{7}"
MD 950 PRINTTAB(10)"{5 SPACES}
B{2 SPACES}{WHT}{3 @}
{7}{2 SPACES}B"
SD 960 PRINTTAB(10)"CCCCC{E}CC
CCCCC{E}CCCC"
MK 970 PRINTTAB(10)"{8 SPACES}
{WHT}{T}↑{T}{7}"
CJ 980 PRINT "{3 DOWN}↑ NOTE: I
F BALANCED O THEN R3 IS
HALVED"
CF 990 GOTO820
SD 1000 REM ***{2 SPACES}DISPL
AY MIN-LOSS{2 SPACES}*
**
AC 1010 PRINT "{CLR}"HDS"
{2 SPACES}"ANS(3)
QC 1020 PRINTTAB(5)CC$(PH)
PJ 1030 PRINTTAB(49)"CCCC{RVS}
{WHT}{3 SPACES}{7}
{OFF}{CC}{R}CCCCCCCC"
MH 1040 PRINTTAB(10)"
{8 SPACES}B"
RG 1050 PRINTTAB(10)"
{3 SPACES}R1{3 SPACES}
{RVS}{WHT} {7}{OFF}"
ES 1060 PRINTTAB(10)"
{8 SPACES}{RVS}{WHT}

```

```

{7}{OFF} R3"
FS 1070 PRINTTAB(10)"
{8 SPACES}{RVS}{WHT}
{7}{OFF}"
FJ 1080 PRINTTAB(10)"
{3 SPACES}{WHT}{3 @}
{7}{2 SPACES}B"
QS 1090 PRINTTAB(9)"CCCCCCCC
{E}CCCCCCCC"
CD 1100 PRINTTAB(10)"
{3 SPACES}{WHT}{T}↑{T}
{7}"
BM 1110 PRINT "{3 DOWN}↑ NOTE:
{SPACE}IF BALANCED THE
N R1 IS HALVED."
SD 1120 PRINT "{DOWN}R1="R1TAB(
30)"R3="R3
KE 1130 GOTO830
KS 1140 REM *** SUB-RTES ***
PQ 1150 POKE781,RO:POKE782,CO:
SYS65520:RETURN
SH 1160 RO=22:CO=0:GOSUB1150:P
RINT "{CYN}1-ENTER NEW
{SPACE}R VALUES
{4 SPACES}2-MENU
{4 SPACES}3-QUIT{7}"
CK 1170 GETA$:IFA$<"1"ANDA$<
"2"ANDA$<"3"THEN1170
RM 1180 IFA$="2"THEN160
XR 1190 IFA$="3"THENPRINT"
{CLR}":POKE53281,6:POK
E53280,14:END
AJ 1200 PRINT "{CLR}":GOSUB1270
:ONVAL (P$)GOTO390,520,
640
AA 1210 PRINT "{DOWN}{CYN}HIT S
HIFT TO REDO{7}":WAIT6
53,1:GOTO160
KH 1220 PRINT "{RVS}VALUE ENTER
ED IS > OR < THAN 20%
{SPACE}OF":PRINTTAB(10
)R:RETURN
AD 1230 PRINT "{DOWN}{RVS}THE I
NSERTION LOSS IS < MIN
.LOSS -"
MK 1240 PRINT "{5 SPACES}{RVS}R
EDO OR CHOOSE MIN.LOSS
":GOTO1210
RR 1250 PRINT "{DOWN}{RVS} SOME
INSERTION LOSS MUST E
XIST":GOTO1210
JK 1260 PRINT "{DOWN}{RVS} SOME
IMPEDANCE MUST EXIST
{SPACE}":RETURN
HP 1270 REM *** GET NEW R VALU
ES ***
MP 1280 RO=4:CO=5:GOSUB1150:PR
INT "{CYN}{RVS}RETURN
{OFF} ONLY FOR ORIGINA
L R VALUE{7}{DOWN}"
RK 1290 PH=2:R1$="":R2$="":R3$
=""
CR 1300 INPUT"ENTER NEW R1 VAL
UE";R1$:R1=VAL(R1$):IF
R1$=""THENR1=RA:GOTO13
20
DK 1310 IFABS (VAL (R1$)-RA)>.2*
RATHENR=RA:R1$="":GOSU

```

```

B1220:GOTO1300
KG 1320 IFP$="3"THEN1350
CA 1330 INPUT"ENTER NEW R2 VAL
UE";R2$:R2=VAL(R2$):IF
R2$=""THENR2=RB:GOTO13
50
SG 1340 IFABS (VAL (R2$)-RB)>.2*
RBTHENR=RB:R2$="":GOSU
B1220:GOTO1330
SS 1350 INPUT"ENTER NEW R3 VAL
UE";R3$:R3=VAL(R3$):IF
R3$=""THENR3=RC:GOTO13
70
XA 1360 IFABS (VAL (R3$)-RC)>.2*
RCTHENR=RC:R3$="":GOSU
B1220:GOTO1350
JR 1370 IFR1=RAANDR2=RBANDR3=R
CTHENPH=1
JG 1380 RETURN

```

Robert C. Marcus lives in St. Thomas, Ontario, Canada. He's the author of CoilCalc (June 1991). □

TYPING AIDS

MLX, our machine language entry program for the 64 and 128, and The Automatic Proofreader are utilities that help you type in Gazette programs without making mistakes. To make room for more programs, we no longer include these labor-saving utilities in every issue, but they can be found on each Gazette Disk and are printed in all issues of Gazette through June 1990.

If you don't have access to a back issue or to one of our disks, write to us, and we'll send you free copies of both of these handy programs. We'll also include instructions on how to type in Gazette programs. Please enclose a self-addressed, stamped envelope.

Write to Typing Aids, COMPUTE's Gazette, 324 West Wendover Avenue, Suite 200, Greensboro, North Carolina 27408.